AFRL-SN-WP-TR-2001-1123

Statistical Pattern Recognition for Synthetic Aperture Radar (SAR)/Automatic Target Recognition (ATR)
Volume 2

Dr. Jian Li
Dr. Jose C. Principe

University of Florida
Department of Electrical and Computer Engineering
437 EB, P.O. Box 116130
Gainesville, FL 32611

JULY 2001

FINAL REPORT FOR PERIOD OF 15 MAY 1999 – 01 JUNE 2001

SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE US GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Louis A. Tamburino

LOUIS A. TAMBURINO
Target Recognition Branch
Project Engineer

Dale E. Nelson

DALE E. NELSON, CHIEF
Target Recognition Branch

Clyde R Heddings

CLYDE R. HEDDINGS, Major, USAF
Deputy, Sensor ATR Technology Division
Sensors Directorate

Do not return copies of this report unless contractual obligations or notice on a specific document requires its return.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** July 2001	**3. REPORT TYPE AND DATES COVERED** Final, 05/15/1999 – 06/01/2001

4. TITLE AND SUBTITLE Statistical Pattern Recognition for Synthetic Aperture Radar (SAR)/Automatic Target Recognition (ATR) Volume 2	**5. FUNDING NUMBERS** C: F33615-99-1-1507 PE: 63762E PR: ARPS TA: NA WU: 0L
6. AUTHOR(S) Dr. Jian Li Dr. Jose C. Principe	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Florida Department of Electrical and Computer Engineering 437 EB, P.O. Box 116130 Gainesville, FL 32611	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) SENSORS DIRECTORATE AIR FORCE RESEARCH LABORATORY AIR FORCE MATERIEL COMMAND WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318 POC: Louis A. Tamburino, AFRL/SNAT, 937-255-1115 x4389	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER** AFRL-SN-WP-TR-2001-1123

11. SUPPLEMENTARY NOTES
The work in this report is published in several journals and conference proceedings. This is Volume 2 of 2.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(Maximum 200 Words)***

State-of-the-art research on spectral estimation, feature extraction, and pattern recognition algorithms are presented for radar signal processing and automatic target recognition. Advanced space-time spectral estimation algorithms are presented for multiple moving target feature extraction as well as clutter and jamming suppression for airborne high range resolution (HRR) phased-array radar. A nonparametric adaptive filtering-based approach, referred to as the Gapped-data Amplitude and Phase EStimation (GAPES) algorithm, is proposed for the spectral analysis of gapped data sequences as well as synthetic aperture radar (SAR) imaging with angle diversity data fusion. A QUasi-parametric ALgorithm for target feature Extraction (QUALE) algorithm is also investigated for angle diversity data fusion. Support Vector Machines (SVMs) as compared with other advanced classifiers in the MSTAR Public Domain Release and HRR data are found to outperform neural networks and matched filters. A new concept to create negative examples from the known target class is presented and shown to tremendously improve the rejection of confusers. Finally, Information Theoretic Learning (ITL) is proposed as a new algorithm to demix HRR signatures of closely parked targets.

14. SUBJECT TERMS Space-Time Processing, Moving Target Detection and Feature Extraction, SAR Imaging, Angle Diversity, Data Fusion, Support Vector Machines, Confuser Rejection, Information Theoretic Learning	**15. NUMBER OF PAGES** 80
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** SAR

Table of Contents

Section **Page**

Abstract iv
Support Vector Machines for SAR/ATR. 1A
I. Introduction 1A
II. Learning Criteria for Empirical Risk Minimization and Structure Risk Minimization 3
2.1 The Perceptron Criterion and the Minimum Squared Error Criterion 3
2.2 Criteria for Structure Risk Minimization 5
III. Experimental Results 10
4.1 Classification Results 13
4.2 Recognition Results 16
VI. SVM Performance in High Range Resolution (HRR) Data 18
VII. Conclusions 20
Mitigation of False Alarms with Negative Examples 22
1. Introduction 22
2. Incorporating Prior Information by Creating Virtual Examples 25
2.1 Creating Virtual Negative Examples 26
2.2 Classifiers 28
3. Experimental Results 29
4 Conclusions 32
Information Theoretic Feature Extraction 33
1. Introduction 33
2. Information Theoretic Learning 35
2.1 Entropy Criterion and Its Applications 36
2.2 Mutual Information Criterion and Its Applications 38
2.3 How Appropriate is the Mutual Information Criterion for Learning? 40
3. Renyi's Entropy 42
4. Quadratic Entropy and Its Nonparametric Estimator 44
5. Quadratic Mutual Information and Cross-Information Potential 47
6. Experimental Results 51
6.1 Blind Source Separation 51
7. Classification 57
8. Conclusion 65
References 67

Abstract

This document presents state-of-the-art research on feature extraction and pattern recognition algorithms for SAR/ATR conducted at the Computational NeuroEngineering Laboratory (CNEL) at the University of Florida with DARPA funding.

We first report on the development and implementation of the newly proposed support vector machine (SVM) for SAR/ATR. We are one of the first groups that proposed such an algorithm and the one that has more extensive experience on the application of SVM to SAR/ATR. We describe here our implementation and an adaptation of the training that avoids the quadratic optimization by working with the dual formulation of the topology. Our implementation makes the SVM even more practical. We compared the SVMs with other advanced classifiers in the MSTAR Public Domain Release and found out that this classifier outperforms neural networks and matched filters. We also apply the SVM to the High Range Resolution (HRR) profiles of MSTAR targets and show the same stellar performance. We conclude by saying that the SVM should be a "must have baseline" for any serious demonstration of classification in SAR/ATR.

The problem of confusers is one of the most serious in ATR. We also present in this report a new concept to create negative examples from the known target classes. We demonstrate with real data that this scheme improves tremendously the rejection to confusers.

The third problem that was researched in this work was the issue of better feature extraction for both representation and classification. We propose information theoretic learning as a new paradigm to project high dimensional data to a smaller subspace while loosing the least of information. We accomplish this by maximizing the output of the system with respect to the desired response. We utilize an algorithm recently developed in the CNEL and apply it to both classification and representation of HRR signatures. The ITL algorithm seems to perform rather well for classification providing classifiers of the same performance as SVM. We also attempted the demixing of HRR signatures of closely parked targets using the ITL algorithm (minimization of mutual information among the outputs of the mapper). Although the preliminary results were positive with artificially mixed data, the tests with XPATCH were inconclusive. More work needs to be done in this difficult problem.

Support Vector Machines for SAR/ATR

I. Introduction

The design of a learning machine is statistical in nature, so an appropriate criterion for the fit is needed between the model and the training set. This implies that the design procedure of the learning machine should take into consideration both the performances of the training set and the model complexity.

In the statistical literature, various criteria for model complexity design have been described, such as the Akaike information-theoretic criterion (AIC) [1], and the minimum description length (MDL) criterion [27,28]. In fact, a common form of the criterion of model complexity can be regarded as a sum of two terms [17,26], i.e., one term of a log-likelihood function, and another term of a model complexity penalty. Generally speaking, the task of a learning machine is to find a weight vector that minimizes the following cost functional $J(w)$ [5],

$$J(\mathrm{w}) = R_{emp}(\mathrm{w}) + \lambda R_{mdl}(\mathrm{w}) \qquad (1)$$

where $R_{emp}(w)$ is the empirical risk or the standard performance measure resulted from the training set, such as the minimum squared error, and the second term $R_{mdl}(w)$ is a complexity penalty term depending on the network topology. In fact, this risk equation (1) is a simple form of regularization theory [33], where λ, the regularization parameter, is normally difficult to compute. When λ is zero, Equation (1) implies the empirical risk minimization (ERM) principle. When λ is increased, more emphasis is put on the complexity penalty to specify the network. This means that a suitable balance should be struck between the accuracy attained on the particular training set, and the capacity of the classifier.

Besides criterion (1), structural risk minimization (SRM) is another inductive principle for learning, which controls the generalization ability of learning machines in the small sample set limit [35]. In the small sample case, Vapnik proposed to minimize the confidence interval, instead of striking the compromise between empirical risk and machine complexity.

In this paper, structural risk minimization will be employed to implement pattern classifiers. The theoretical and experimental results show that many learning algorithms, such as SVMs [35], AdaBoost [10, 31], and Bagging [3], will produce classifiers with large margins and lead to better generalization performance. As a large margin classifier, the SVM has been used successfully in many pattern recognition applications [5], including isolated handwritten digit recognition [6], automatic target recognition [40], speaker identification [32], face detection in images, and text categorization.

In most pattern classification applications, one needs to perform the classification into a fixed number of classes, where a close or pre-determined set of classes is usually specified in advance. However, generalization is not the only problem in real world applications. In some practical cases, some exemplars presented to the classifier during testing do NOT belong to the learned classes. For instance, in face recognition, a security system has to be able to reject intruders while being able to cope with variations of a known face due to lighting or pose differences [4]. In automatic target recognition, the system should be able to discriminate between military and civilian vehicles [23]. But since it is impossible to create a training set with all possible vehicles, this class of problems has been called classification of open sets or recognition. Similar problems arise in speaker identification [14], recognition and verification of fingerprints, signatures, etc. This is an important problem that falls between classification as we normally use it (i.e. all the test exemplars belong to one of the classes) and detection [18].

One common way of implementing rejection is the thresholding criterion, which defines a decision region in the pattern space with a threshold T given in advance as,

$$D(T) = \{x \mid g(x) \geq T, \forall x \in R(I)\} \qquad (2)$$

where $g(x)$ is the decision function of a classifier, and $R(I)$ represents the pattern space. In many ways this is similar to the Neyman-Pearson criterion [18] where the detector's figure of merit (probability of detection versus false alarms) as a function of threshold has to be plotted to find the actual performance. The thresholding criterion is illustrated in Figure 1, where one can easily realize that the locality of the discriminant function is a fundamental requirement for a verification system. This implies that the classifier should be able to create a "local" decision region, instead of a "global" one, otherwise a confuser far away from the class center can easily be accepted as an object of interest.

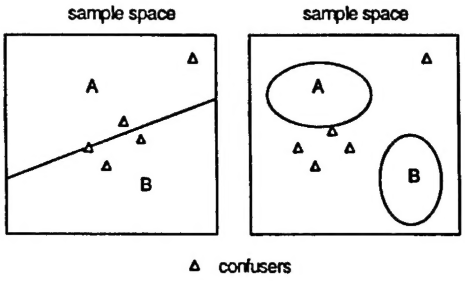


Figure 1 An illustration of a two-class classification problem. In the left figure, a "global" discriminant function divides the whole sample space into two parts. In the right figure, two "local" decision regions are formed to keep the confusers away from the class region of interest.

For the problem of SAR/ATR, the classifier should be able to classify the targets in the training set as well as their variants (different serial numbers), and to reject confusers, all at a reasonable level. Confusers in this paper are vehicles not included in the training set. In this paper, SVMs are utilized to perform the task of target recognition and confuser rejection. As a comparison, the perceptron trained with the minimum squared error criterion (the delta rule) [17] is employed to perform the same tasks. The theoretical background of MSE and structural risk minimization are given in Section II. Experimental results and discussion are given in Section III and IV, respectively.

II. Learning Criteria for Empirical Risk Minimization and Structure Risk Minimization

Let us consider a two-class classification problem, where the training set is described as $X := \{\mathbf{x}_1,,...,\mathbf{x}_m\}$, $\mathbf{x}_i \in R^n$, and labels $Y := \{y_1,,...,y_m\} \subseteq \{-1,1\}$.

2.1 The Perceptron Criterion and the Minimum Squared Error Criterion

A simple classifier that can solve a linearly separable task is the perceptron [29], and its linear decision function is represented as

$$g(x) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b) \qquad (5)$$

where $(\mathbf{w} \cdot \mathbf{x})$ indicates the inner product, and *sgn*(.) is a signum function.

The algorithm used to adjust the parameters w and b of this model first appeared in a learning procedure developed by Rosenblatt [29] for a brain model. The perceptron criterion function (or the risk functional) is defined as

$$J(w) = \sum_{\mathbf{x}_i \in E_x} |\mathbf{w} \cdot \mathbf{x}_i + b| \qquad (6)$$

where the summation is over the set of E_x of patterns that are mis-classified by the perceptron, and $|\cdot|$ represents the absolute value. It was proven that, for linearly separable problems, the algorithm converges in a finite number of iteration [17].

However, when the training set is not linearly separable, the solution of separating the training data with the smallest number of errors is NP-complete. Moreover, the gradient based algorithm cannot be applied to find a minimum of the cost function since for the cost function (6), the gradient is either zero or undefined. Unlike the perceptron criterion (6) which considers only the mis-classified patterns, the minimum squared error criterion takes into account the entire training set, which is defined as the squared error (L_2 norm) between the desired output and actual output,

$$J(w) = \sum_{i=1}^{m} (y_i - g(w, \mathrm{x}_i))^2 \tag{7}$$

To get a continuous differential output, a sigmoid function is used,

$$g(w, x) = \frac{1}{1 + \exp(-w \cdot x + b)} \tag{8}$$

Then the delta rule which is a gradient based algorithm can be used to train the network. Since the samples that produce larger errors are closer to the boundary, the MSE risk functional (7) will place the decision surface at a location that predicts better the correct side of threshold than the perceptron criterion does, i.e., provides a "large" margin between classes.

Besides the empirical risk (the error of training set), one can also take into consideration the model complexity using regularization theory as indicated in Equation (1). In this paper, the following cost functional form with regularization term is applied,

$$J(\mathrm{w}) = \sum_{i} (y_i - g(\mathrm{w}, \mathrm{x}_i))^2 + \lambda \cdot \|\mathrm{w}\|^2 \tag{9}$$

The well known weight elimination [37] procedure refers to the minimization of this functional.

2.2 Criteria for Structure Risk Minimization

The perceptron trained with equation (9) implements criterion (1). However, as indicated in [19], neither the perceptron criterion nor the MSE criterion would necessarily lead to a minimum classification error, i.e., good generalization ability. In this section, a new learning criterion for structural risk minimization [35] is considered. Two applications of this learning methodology, the optimal hyperplane and the SVM, are introduced.

(1). The Optimal Hyperplane

The training set of Section II is said to be separated by an Optimal Hyperplane if the following two conditions are satisfied. First, all the samples are separated without error (keep the empirical risk zero), and second, as illustrated in Figure 2, the distances between the closest vectors to the hyperplane are maximal. The separating hyperplane is described in the canonical form, i.e.,

$$\begin{aligned} &\mathrm{w} \cdot \mathrm{x}_\mathrm{i} + b \geq 1 && \textit{if } \; y_i = 1 \\ &\mathrm{w} \cdot \mathrm{x}_\mathrm{i} + b \leq -1 && \textit{if } \; y_i = -1 \end{aligned} \qquad (10)$$

In a more compact form, the following notation is used,

$$y_i(\mathrm{w} \cdot \mathrm{x}_i + b) \geq 1, \;\; i = 1, \ldots, m \qquad (11)$$

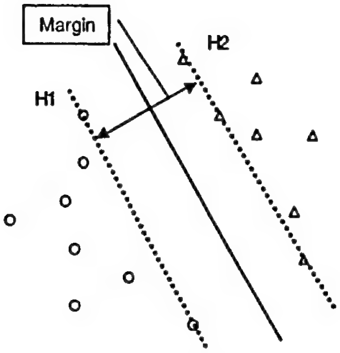


Figure 2. A two-class linearly separable problem (balls vs. triangles). The optimal hyperplane (solid line) intersects itself halfway between the two classes, and keeps the margin maximal. The samples across the boundary *H1* or *H2* are support vectors.

It is easy to prove that the margin between the two hyperplanes $H_1: \mathbf{w}\cdot\mathbf{x}_i + b = 1$ and $H_2: \mathrm{w}\cdot\mathrm{x}_i + b = -1$ is $d = 2/\|\mathrm{w}\|$. Thus, to find a hyperplane that satisfies the second condition, one has to solve the quadratic programming problem of minimizing $\|\mathrm{w}\|^2$, subject to constraint (11). The solution to this optimization problem is given by the saddle point of a primal Lagrange functional,

$$L_P = \frac{1}{2}\|\mathrm{w}\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathrm{w}\cdot\mathrm{x}_i + b) - 1] \qquad (12)$$

where α_i, $i = 1,\dots,m$, are positive Lagrange multipliers. Since (12) is a convex quadratic programming problem, this means that it is equivalent to solve a "dual" problem [7]: maximize L_P, subject to the constraints that the gradient of L_P with respect to *w* and b vanish, which gives the conditions,

$$\mathrm{w} = \sum_i \alpha_i y_i \mathrm{x}_i \qquad (13)$$

$$\sum_i \alpha_i y_i = 0 \qquad (14)$$

Substituting (13) and (14) into (12), we get the dual problem of maximizing,

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathrm{x}_i \cdot \mathrm{x}_j$$
$$s.t.\ \ \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

Also, we can use the following vector representation,

$$\begin{aligned} &L_D = \ddot{\mathrm{E}}^T 1 - \frac{1}{2}\ddot{\mathrm{E}}^T \mathrm{C}\ddot{\mathrm{E}} \\ &s.t.\ \ \ddot{\mathrm{E}}^T \mathrm{Y} = 0 \\ &\qquad \ddot{\mathrm{E}} \geq 0 \end{aligned} \tag{15}$$

where $\ddot{\mathrm{E}}^T = (\alpha_1,...,\alpha_m)$ is a parameter vector, $1^T = (1,...,m)$ is an m-dimensional unit vector, $\mathrm{Y}^T = (y_1,..., y_m)$ is the m-dimensional label vector, and **C** is a symmetric m by m correlation matrix with elements $C_{ij} = y_i y_j \mathrm{x}_i \cdot \mathrm{x}_j, i, j = 1,...,m$. Notice that there is a Lagrange multiplier α_i for every training sample. In the solution, those points for which $\alpha_i > 0$ are called "support vectors" (SV), and lie on either H_1 or H_2. The separating rule is, based on the Optimal Hyperplane,

$$g(x) = \mathrm{sgn}(\sum_{i \in SV} y_i \alpha_i \mathrm{x}_i \cdot \mathrm{x} + b) \tag{16}$$

(2). The Soft Margin Hyperplane

More generally, when dealing with non-linearly separable patterns, we will introduce positive slack variables ξ_i, $i = 1,....,m$, in the constraint (10), i.e.,

$$\begin{aligned} &\mathrm{w} \cdot \mathrm{x_i} + b \geq 1 - \xi_i && \text{if } y_i = 1 \\ &\mathrm{w} \cdot \mathrm{x_i} + b \leq -1 + \xi_i && \text{if } y_i = -1 \\ &\qquad \xi_i \geq 0 \ \ \forall i \end{aligned} \tag{17}$$

For an error to occur, the corresponding ξ_i must exceed unity, thus $\sum_i \xi_i$ is an upper bound on the number of training errors. In this case, the risk functional that we want to minimize is,

$$L = \|w\|^2/2 + \lambda(\textstyle\sum_i \xi_i^\sigma)^k \qquad (18)$$

subject to (17), where λ is a parameter to assign a penalty to training errors. For any positive integer k, this is a convex programming problem. For sufficiently large λ and sufficiently small σ, the parameters **w** and bias b determine the hyperplane that minimizes the number of errors on the training set and separate the rest of the elements with maximal margin. Note that the problem of constructing a hyperplane which minimizes the error on the training set is general *NP*-complete. To avoid this difficulty the case of σ=1 is considered in this paper, where the solution is called the ***soft margin*** hyperplanes. If we take k=2 in (18), it remains a quadratic programming problem of maximizing [5],

$$\begin{aligned} &L_D = \text{Ë}^T 1 - \frac{1}{2}\left[\text{Ë}^T \text{CË} + \frac{\delta^2}{\lambda}\right] \\ &s.t.\ \ \text{Ë}^T \text{Y} = 0 \\ &\qquad \delta \geq 0 \\ &\qquad 0 \leq \text{Ë} \leq \delta 1 \end{aligned}$$

where δ is a scalar. If we take $\delta = \alpha_{\max} = \max(\alpha_1, ..., \alpha_m)$, then the problem is a convex programming problem of maximizing,

$$\begin{aligned} &L_D = \text{Ë}^T 1 - \frac{1}{2}\left[\text{Ë}^T \text{CË} + \frac{\alpha_{\max}^2}{\lambda}\right] \\ &s.t.\ \ \text{Ë}^T \text{Y} = 0 \\ &\qquad \text{Ë} \geq 0 \end{aligned}$$

Therefore, to construct a soft margin hyperplane, one can either solve convex programming problem in the m dimensional space of the parameter vector **Ë**, or solve the quadratic programming problem in the dual m+1 space of Ë and δ [5].

(3). Support Vector Machine

Until now, all the previous architectures create the decision functions that are all linear functions of data. Then one may ask how can the above method be generalized to the case of a nonlinear decision function? One alternative is to map the data to some other high dimensional (possibly infinite dimensional) Euclidean space (feature space) using a mapping $\phi : R^d \to E$. There is evidence provided by Cover's theorem [13] that a complex pattern classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space. The advantage of this method is that it decouples the numbers of free parameters of the learning machines from the input space dimensionality [18]. In this way, the decision rule of (16) is implemented in the new feature space, i.e.,

$$g(x) = \text{sgn}(\sum_{i \in SV} y_i \alpha_i \phi(x_i) \cdot \phi(x) + b)$$

By the Mercer's condition [6, 28], there exists a mapping ϕ and a symmetric function *K(x,y)* which has an expansion $K(x, y) = \sum_{k=1}^{\infty} \phi_k(x)\phi_k(y)$, if and only if, for any *f(x)* such that $\int f^2(x)dx$ is finite, there exists,

$$\int K(x, y) f(x) f(y) dxdy \geq 0$$

The convolution of the inner product allows the construction of a decision function that is nonlinear in the input space,

$$g(x) = \text{sgn}(\sum_{i \in SV} y_i \alpha_i K(\mathrm{x}_i, \mathrm{x}) + b) \qquad (19)$$

and this is also equivalent to a linear decision function in the high-dimensional feature space of $\phi_1(x),..., \phi_m(x)$. This learning machine is the so-called Support Vector

Machine. Correspondingly, the task of this quadratic programming problem is to maximize,

$$\begin{aligned} & L_D = \ddot{\mathrm{E}}^T 1 - \frac{1}{2}\left[\ddot{\mathrm{E}}^T \mathrm{K} \ddot{\mathrm{E}} + \frac{\delta^2}{\lambda}\right] \\ & s.t.\ \ \ddot{\mathrm{E}}^T \mathrm{Y} = 0 \\ & \qquad \delta \geq 0 \\ & \qquad 0 \leq \ddot{\mathrm{E}} \leq \delta 1 \end{aligned} \tag{20}$$

where **K** is a symmetric m by m kernel matrix with elements.

To describe the classification ability of either the Optimal hyperplane or the SVM, the margin of an example ($\mathbf{x}_j, y_j$) is defined as

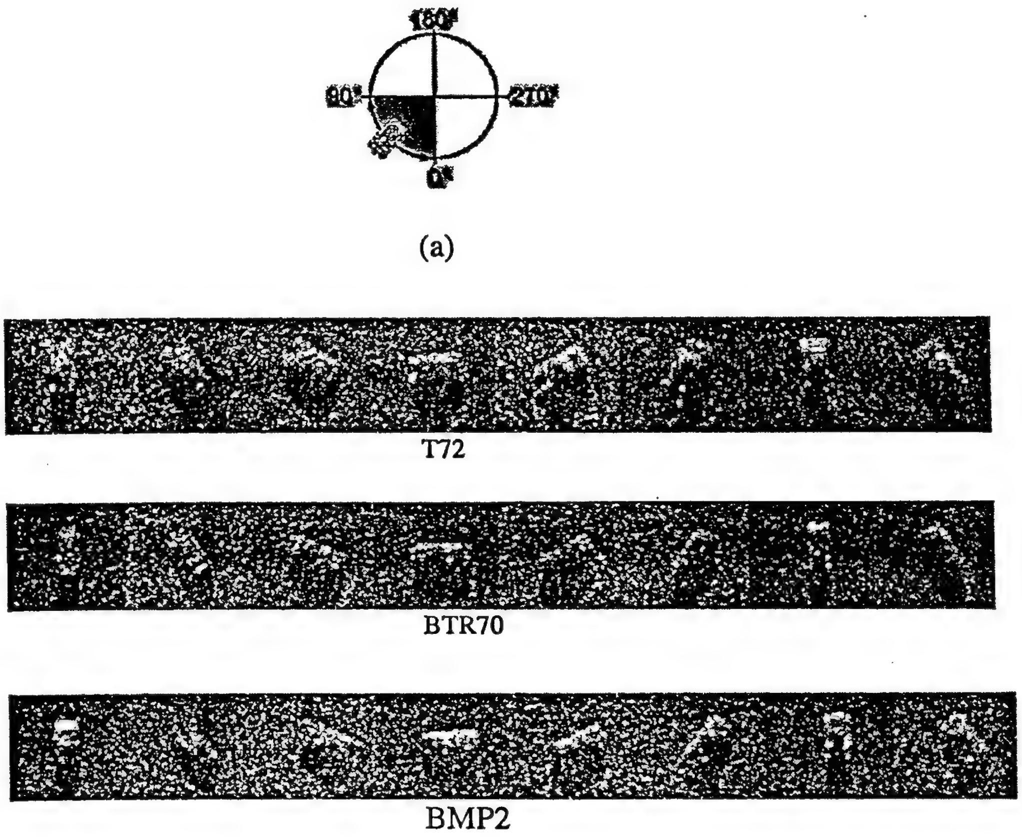
$$\rho_f(\mathrm{x}_j, y_j) = y_j g(\mathrm{x}_j) \tag{21}$$

It will be observed in the following experiments that SVM tends to increase the margins associated with examples and converge to a distribution in which most examples have large margins.

III. Experimental results

Automatic target recognition (ATR) generally refers to the use of computer processing to detect and recognize target signatures in sensor data. The conventional ATR architecture comprises a focus of attention (detector and discriminator) followed by a classifier [24]. The role of the focus of attention is to discard image chips that do not contain potential targets. ATR classifiers can be broadly divided into two types following the taxonomy in [20]: one class in one network (OCON) and all class in one network (ACON). Template matching [36] is typical in the OCON group, while some discriminant classifiers such as the multi-layer perceptron (MLP) or radial basis function networks [38] appear in the second class.

In this paper, synthetic aperture radar (SAR) automatic target recognition (ATR) experiments were performed using the MSTAR database to classify three targets and reject confusers. The data are 80 by 80 SAR images drawn from three types of ground vehicles: the T72, BTR70, and BMP2 as shown in Figure 3. These images are a subset of the 9/95 MSTAR Public Release Data [36], where the pose (aspect angles) of the vehicles lies between 0 to 180 degrees. Only images of the vehicles are used here (there is no need for the focus of attention) so they will be directly scored by the classifier.



(a)

T72

BTR70

BMP2

Figure 3. (a) Illustration of pose; (b) SAR images of target T72, BTR70 and BMP2 taken at different aspect angles

The SAR images are very noisy due to the image formation and lack resolution due to the radar wavelength, which makes the classification of SAR vehicles a non-trivial problem [23]. Unlike the optical images, the SAR images of the same target taken at different aspect angles are not very similar with each other, which precludes the existence of a rotation invariant transform. This results from the fact that a SAR image reflects the fine target structure (point scatter distribution on the target surface) at a certain pose. Parts of the target structure will be occluded when illuminated by the radar from another pose, which results in dramatic differences from image to image with angular increments as small as 10 degrees. Model based approaches are being investigated in MSTAR literature, but here we will concentrate on comparing statistical classifiers.

To decrease the complexity of the classification problem, the input space was divided using the pose information [25] and six sub-classifiers were trained, each of which covered approximately thirty degrees of aspect angles (see Figure 4). The rationale behind this classifier architecture is to simplify the bank of matched filters. The matched filter is linear so its generalization is poor, requiring many finely spaced (say, 10 degrees) templates. This on the other hand imposes stringent constraints on the pose estimator, i.e. the pose has to be indicated with an error below the template increment.

We decided to utilize more powerful and robust classifiers, which will generalize better. Presently we are experimenting with 30 degrees sectors. If we are successful then fewer classifiers are needed to cover the full 0-360 degree, and the pose estimator requirements can also be relaxed. Following this reasoning, we have created a pose estimator based on mutual information that is able to determine the pose of all MSTAR targets with an error less than 10 degrees [39]. One of the advantages of this pose estimator is that it is trained off-line, so on-line it is as simple as a memory lookup. The results presented in this paper will be based on the architecture of Figure 4 with the goal of testing its performance against the matched filter filter-bank of Velten [36].

The training set contained SAR images taken at a depression angle of seventeen degrees, while the testing set depression angle is fifteen degrees. So the SAR images between the training and the testing sets for the same vehicle at the same pose are different, which helps to test the classifier generalization. Variants (different serial number) of the three targets were also used in the testing set. The size of training and testing sets is 406 and 724, respectively.

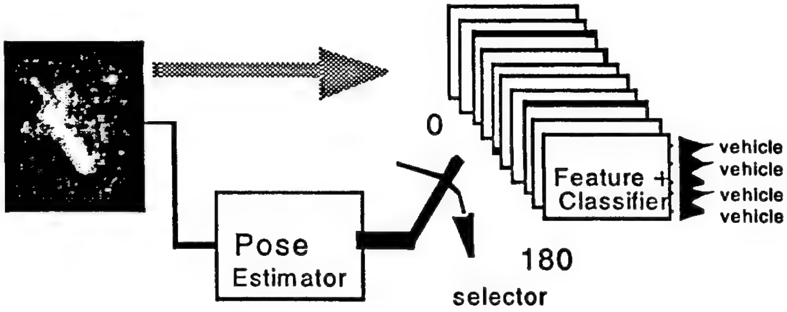
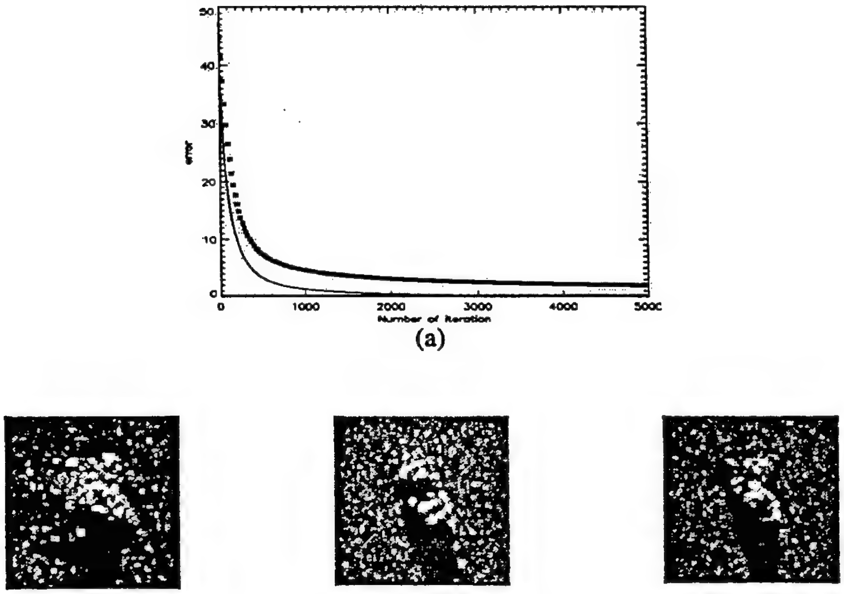


Figure 4 The classifier topology is depicted. First a pose estimator is applied to the image and determines the approximate pose of the target, then a classifier is chosen according to the result of pose estimation.

In the experiment, three classifiers are employed, (1) A perceptron trained with the delta rule, with a single layer structure of 6,400 input units and 3 output units. (2) The optimal hyperplane (OH) classifier (the same perceptron as in (1) but trained with the SRM criterion). (3) The Support Vector Machine, where the Gaussian kernel was employed with the kernel size chosen as the average Euclidean distance between training patterns. Both the OH classifier and the SVM were trained with the kernel Adatron with bias and soft margin algorithm [11,12].

4.1 Classification Results

A classification experiment was performed first. When training the perceptron, one will usually meet a very common problem, i.e. over-training of the network. To solve this problem weight elimination was utilized in training [37]. Figure 5(a) depicts the learning curves for the perceptron. As the training error decreased, the cross-validation error also decreased which implied that no over-training occurred during training. We stopped training at 5,000 iterations. Figure 5(b) depicts images of the input weight matrices for each of the three output nodes for one of the sector classifiers (zero to thirty degrees aspect angles). Compared with the SAR images in Figure 3, one can see that these weights resemble the target images showing that the delta rule optimally scales each image to create a discriminant template for the class.



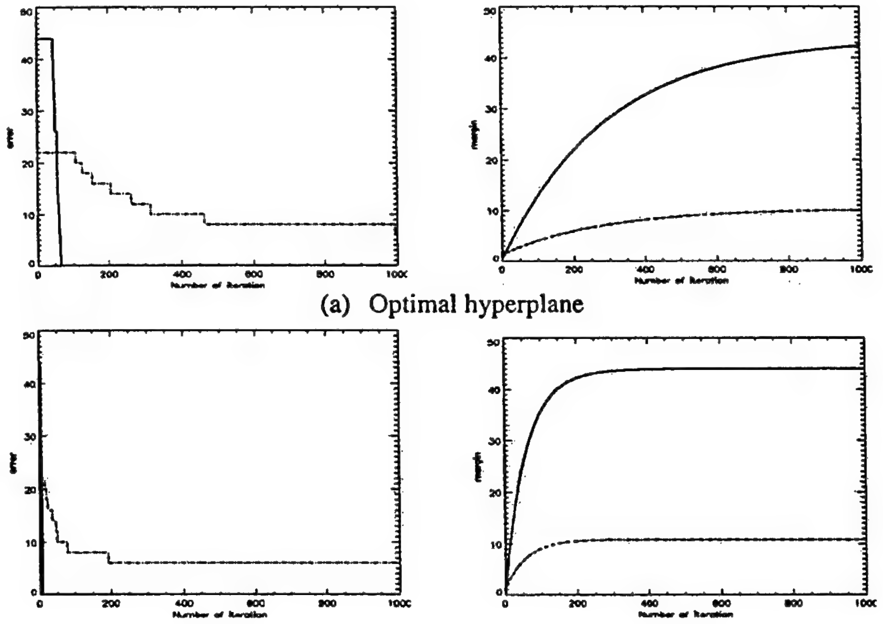
(a)

(b)

Figure 5 (a) Learning curves of the perceptron trained with Equation (9), the solid line is for training set, and the asterisks for cross-validation set; (b) weights connecting input nodes with three output nodes, respectively, of the classifier covering aspect angles from zero to thirty degrees. Compared with the images in Figure 3, they resemble the targets' features.

Figure 6 illustrates the learning curves and margin distribution for the OH classifier and SVM, where the margin distribution graph is defined as the sum of the margins

(Equation (21)) of the training set as a function of number of iterations. In Figure 6(a), the learning curve reveals that the training error dropped to zero in 70 iterations, but the testing error continued to drop from 22 to 8 in the next 450 iterations. In Figure 6(b) the error only took about 20 iterations to reach zero while the testing error continued to drop from 22 to 6 in the following 200 iterations. Meanwhile, the sum of margins of the training set continued to increase quickly even after the training error was zero. And the sum of margins of the testing set also increased slowly after the testing error stopped decreasing.



(a) Optimal hyperplane

(b) Support Vector Machine

Figure 6. Learning curves and margin distribution graphs for the Optimal Hyperplane (OH) and SVM, where the margin distribution graph is defined as the sum of the margins of the training set as a function of number of iterations. The learning curves are shown above the corresponding margin distribution graphs. Each learning curve and margin distribution graph shows the training error/margin (with solid line) and testing error/margin (with dash-dot line), respectively. It is revealed that after the training error dropped to zero, the testing error still continued dropping, and the sum of margins continued increasing.

Table 1 shows the classification results (classification error rate) for the 3 classification methods. It reveals that the OH and the SVM had a slightly better classification performance than the perceptron. Their classification error rates P_e's were around 5% while the perceptron achieved 9% approximately. The networks were run several times with different initial conditions and learning rates and the results of Table I were repeatable.

Table 1 Classification error rates (%) of the classifiers

	BMP2	BTR70	T72	Average
Perceptron	9.35	0.93	11.4	8.98
OH	6.45	1.87	5.28	5.25
SVM	7.74	0.93	4.56	5.39

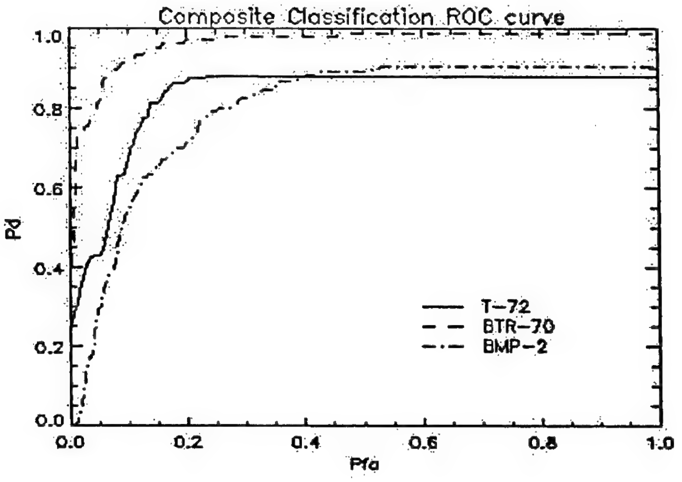
4.2. Recognition Results

A critical problem in ATR is how to discriminate between targets and confusers. When we cannot guarantee that all the vehicles found in the test set belong to the training set classes, rejecting patterns with a low degree of membership to these classes becomes important.

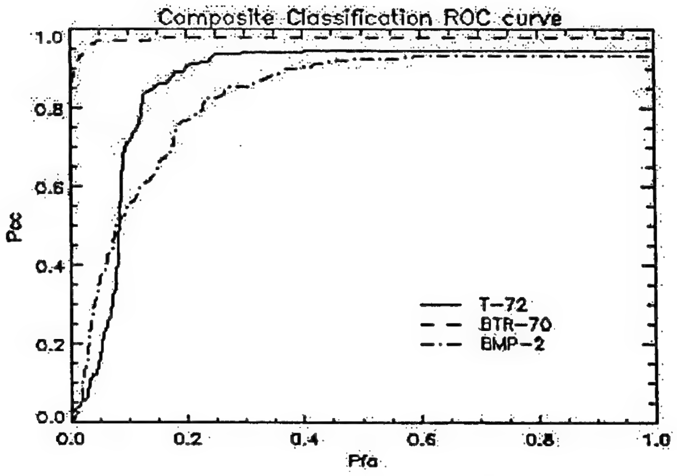
In the recognition experiment two confusers, D7 and 2S1, were added to the testing set. The recognition results are listed in Table 2, where a threshold was set to keep the probability of detection P_d in the testing set equal to 0.9. This setting was chosen to provide direct comparison with a baseline study done on the same data with template matchers [36]. It is shown that the three classifiers gave very different recognition performances. The recognition error rates P_e's of the OH classifier, SVM and the perceptron were 2.76% and 2.07%, and 4.56%, respectively. These values are lower than the ones in Table I since all the outputs below threshold are considered rejections and appear in the confuser column. When confusers were added to the testing set, the SVM

Table 2 The recognition error rates (%) of the classifiers and the false alarm rates (%) with respect to the confusers

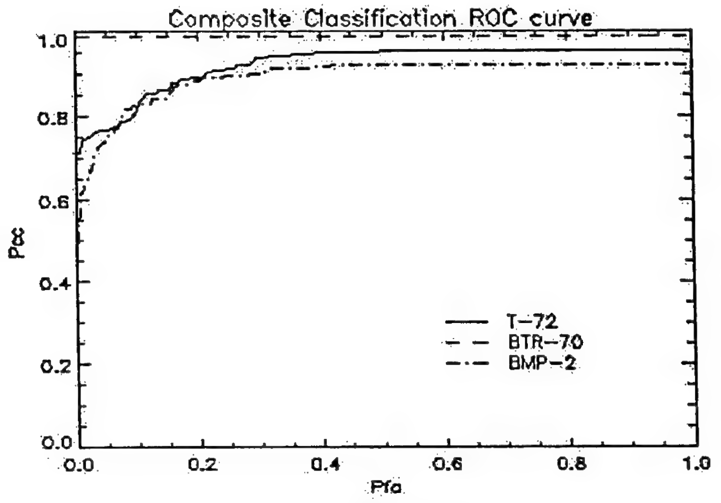
	BMP2	BTR70	T72	Average	Confuser rejection
Perceptron	3.87	1.87	6.19	4.56	21.82
OH	3.87	0.93	2.28	2.76	48.00
SVM	3.55	0.93	0.98	2.07	67.64



(a) perceptron



(b) Optimal Hyperplane



(c) SVM

Figure 7. ROC curve of the three classifiers, using the test sets against the two confusers.

showed the highest rejection rate of 67.64%, while the optimal hyperplane presented a rejection rate of 48%, and the perceptron 21.82%, respectively.

To give a overall performance comparison, the receiver operating characteristics (ROC) curve of the three classifiers is shown in Figure 7. It is observed that the SVM shows much better target recognition and confuser rejection performance than the two other classifiers, the OH and perceptron.

VI- SVM performance in High Range Resolution (HRR) data.

We compared a template matcher, a neural network classifier and a Support Vector Machine (SVM) in HRR data obtained from the HRR Public Released Public Target HRR DATA CDROM. Four targets were used (Btr70_c71, t72_132, 2s1_b01, zi1131_b01). We added white noise so that the SNR is 15 db.

We utilized two different spectral estimators to reconstruct the HRR profiles to compare their effect in detection accuracy. First, we implemented an FFT based method to obtain the Range Profile (128 sample long) from the phase history. We also utilized MRELAX to get the features (16 scatters) then take FFT to obtain range profile again (also of 128 samples). The training and test sets had each 236 exemplars.

The template matcher was implemented by the normalized correlation between the test y and each of the train profiles x_i to find the best match

$$C_i = \frac{\left|x_i^H y\right|}{\sqrt{\left|y^H y\right|\left|x_i^H x_i\right|}}$$

The neural network classifier was a single layer perceptron with 128 inputs and 4 outputs. It wastrained with the backpropagation algorithm as before. We used this simple topology due to the lack of training data. With more data other more sophisticated topologies could be used to improve the results, but to guarantee good generalization we had to restrict our ANN topology.

The Support Vector Machine (SVM) utilized the Gaussian Kernel and the Adatron algorithm for training. A detailed description of this classifier was presented in the previous sections. Table I shows the results of our tests.

Table III Miscalssification error

	Correlation	NN	SVM
MFFT	35	21	9
MRELAX	34	18	8

As we can observe from the Table, the SVM is the best classifier with a misclassfication error of only 8-9%, while the template matcher produced 34-35%. The neural network is able to capture a bit more of the target structure, with a misclassification error of 18 – 21%, but still wait above the results for the SVM. The next table shows the confusion matrix for the SVM machine in the test set.

Table IV. Confusion matrix for the SVM classification of HRR profiles

class	1	2	3	4
1	57	1	0	1
2	4	58	1	3
3	0	3	37	0
4	1	3	0	71

We also verified that the Mrelax does not improve significantly the classification accuracy, although it improves substantially the detail of the image reconstruction. Our conclusion is that the SVM seems to be a viable classifier not only for SAR/ATR but also for HRR profiles. Further analysis is necessary, since this training and test sets are small.

VII. Conclusions

This workshows that the SVM provides a new approach to the problem of automatic recognition. Comparison between SVM and the perceptron shows that the SVM presents a good target recognition performance. Moreover, the SVM with Gaussian kernel functions is able to form a local or "bounded" decision region that presents better rejection to confusers. Another advantage is that, given a small sample size problem, the freedom of the classifier trained by SRM criterion is much smaller than that trained with the MSE criterion.

Mitigation of False Alarms with Negative Examples

1 Introduction

The problem of learning from examples in pattern recognition can be modeled as approximating some unknown target function g with some hypothesis class H_n and a training set $D = \{(x_i, y_i), i = 1, ..., m\}$, such that a cost function $J(y_i, g(x_i))$ is minimized, where the $x_i \in R^n$ is a training example from a n-dimensional space and y_i is the corresponding label. Based on the learning theory of pattern recognition, the generalization error consists of two components: an approximation error and an estimation error.

$$J = J_{app}(error) + J_{est}(error) \tag{1}$$

To make the approximation error small, one need more complex (or, large-sized) models; to make the estimation error small one need less complex (small-sized) models. This compromise between the approximation error and estimation error arises in all machine learning methods (Barron, 1994; Niyogi et al, 1996).

A possible solution to this problem is to use noisy examples in the training or adding noise to the training set to improve generalization (Holmstrom & Koistinen, 1992;Webb,1994). A more efficient solution is to utilize prior information about the target function g, which reduces the size of the target class and helps solving the problem of poor generalization (Niyogi et al, 1998). One technique is based on *hints* (Abu-Mostafa, 1995), where *hints* are the auxiliary information about the target function that can be used to guide the learning process. Another technique is to use the prior knowledge in the design of the learning rules. The tangent distance proposed by (Simard, 1993) is one of the best examples of this approach.

The other way of incorporating prior knowledge about the target function g is to generate novel examples from the known training set, thereby enlarging the effective data set. These additional examples, created from the existing ones by the application of prior knowledge, are the so-called virtual examples (Niyogi et al 1998; Abu-Mostafa, 1995). Suppose that we have prior knowledge of a set of transformations that can be used to obtain new examples from old. Given the training set $D = \{(x_i, y_i), i = 1, ..., m\}$ and knowledge of some transformation T, the virtual example set $D' = \{(x'_i, y'_i), i = 1, ..., m\}$ can be obtained by

$$(x_i, y_i) \xrightarrow{T} (x'_i, y'_i) \qquad (2)$$

As discussed above, the problem of learning from examples can be modeled as one of function approximation and can be formulated in the framework of regularization theory (Tikhonov & Arsenin,1977). In this framework the solution is found by minimizing a functional of the form

$$H[g] = \sum_{i=1}^{m} (g(x_i) - y_i)^2 + \lambda \|Pg\|^2 \qquad (3)$$

where λ is a positive regularization parameter, P is a differential operator, and $\|Pg\|^2$ is a cost functional that constraints the space of possible solutions according to some form of prior knowledge, e.g., smoothness. According to the regularization theory, the solution of this problem has the following form

$$g(x) = \sum_{i=1}^{m} \alpha_i K(x, x_i) + b \qquad (4)$$

where α_i and b are the corresponding coefficients of kernel functions $K(x)$ and bias, respectively.

In many situations, further information about a function may consist in knowing that its value at some points has to be far from a predetermined value. This is the problem of learning from positive and negative examples. The *positive examples* represent the points that the function ought to be close to, while the *negatives examples* are those regions that the function must avoid. Girosi (Girosi et al, 1991) showed ways of dealing with learning in presence of unreliable examples or outliers. Suppose that we create a set of virtual negative examples $D' = \{(x'_i, y'_i), i = 1, ..., m\}$, then the functional (3) becomes(Girosi et al., 1991)

$$H[g] = \sum_{i=1}^{m}(g(x_i) - y_i)^2 - \sum_{i=1}^{m}(g(x_i') - y_i')^2 + \lambda \|Pg\|^2 \tag{5}$$

where the second term corresponds to the negative examples, and the solution becomes

$$g(x) = \sum_{i=1}^{m} \alpha_i K(x, x_i) + \sum_{i=1}^{m} \alpha_{i'} K(x, x_i') + b \tag{6}$$

where $\alpha_{i'}$ is the coefficient corresponding to a virtual negative example.

In most pattern classification applications, one needs to perform the classification into a fixed number of classes. However, in some practical cases, some exemplars presented to the classifier during testing do not belong to the learned classes. These are called *confusers* or *intruders*, and the classification problem becomes recognition. In recognition problems the universe of possible inputs is larger than the union of the known classes. The goal of recognition is still to accurately classify the known classes but also reject the confusers. It is clear that the problem became much more complex and that there is an intrinsic conflict between generalization and rejection of confusers. One common way of rejecting confusers is the thresholding criterion (Nilson, 1965). Although the thresholding criterion is able to form a reasonable decision boundary for recognition, the shape of the decision function is actually only determined by the training set and the classifier topology.

When the training set is of insufficient size, the idea of virtual examples can also be applied to recognition, but virtual positive examples only provide a better characterization of a subset of the input space (covered by the classes). In recognition, we also wish to provide information to the classifier that there are regions in the space where the output should be low. This can never be done with positive examples, hence we investigate in this paper the use of negative examples to improve rejection to confusers.

2 Incorporating Prior information by Creating Virtual Examples

Here we will be extending Girosi's ideas to the case of recognition. As we stated above, in recognition the training set spans only part of the space of possible inputs. For instance, in automatic target recognition (ATR) (Novak et al. 1997; Zhao & Principe, 1999), the goal is to create classifiers for specific types of military vehicles from a training set. However, during operation, the ATR system will have to discriminate between military and civilian vehicles. Civilian vehicles have reflection properties very close to military vehicles differing only in the detail, so they are accepted as potential targets by the focus of attention stage of ATR systems (Principe et al.,1998). Since it is impossible to create a training set with all possible civilian vehicles, the ATR system has the very difficult task of not only classifying the military targets but also rejecting the civilian vehicles (called confusers). Similar problems arise in speaker identification, fingerprints recognition or verification, etc.

One common way of rejecting confusers is to create template matchers for the training classes and using the thresholding criterion, which defines a decision region in the pattern space with a threshold T given in advance as

$$D_T = [x | g(x) \geq T, \forall x \in R(I)] \tag{7}$$

where $g(x)$ is the discriminant function of a classifier, and $R(I)$ represents the input space. The idea is to create a membership metric in the output space given by the value of the correlation to the class template. Since template matchers are distance classifiers (Nilson,1965), the thresholding criterion creates a local tessellation around the template such that a confuser is rejected by being far away from the class center. This is the reason why in recognition the figure of merit of a classifier is given by the receiver operating characteristics (ROC) curve (Helstrom,1968), that is, the probability of detection as a function of the number of false alarms. There is an obvious compromise between the detectability of class objects and the rejection of confusers. In fact, small variations between the class template and the object produce a decrease in correlation. Thus,the classifier performance drops with high thresholds because some of the class objects are scored as confusers.

When discriminant classifiers such as Bayes or neural networks are used, the problem becomes compounded by the nonlinear nature of the discriminant functions. Depending upon the classifier topology the discriminant functions can be global (such as Perceptron) or local (such as SVM). One can still use the idea of using the thresholding criterion at the output of the trained classifier, but now we are implicitly using the discriminant function as a proxy to measure the distance to the class, which may be erroneous.

In this context, we can see the importance of virtual negative examples. The exemplars are confined in clusters, and there are large areas not covered by any exemplar. So we wish to populate the input space away from the class centers with exemplars that would train the classifiers to produce low values so that the discriminant functions become better positioned. This should improve the rejection to confusers.

2.1 Creating virtual negative examples

The problem is that we do not have explicitly negative exemplars, that is, we wish to create negative exemplars from the class samples. Each problem may require a different procedure to create virtual negative exemplars, here we propose to apply a maximum entropy (ME) formulation of virtual negative training set.

In SAR ATR the vehicles are recognized by the relative location and number of point scatters, that is, the bright spots in the image. Each vehicle of interest has its own intrisinc structure and reflective characteristics from SAR. Since we have no knowledge of what signatures the confusers (e.g.,civilian vehicles) might have, one thing we can do is to simulate or estimate the statistical signatures, for example, probability density function (pdf) of the confusers from a training set. In SAR ATR, confusers usually have similar features with the targets of interest. Actually, it is hard to get a accurate estimation of pdf since we do not have enough training data. Instead, we propose to apply a distortion function to the class exemplars. We experimented with different distortions, but the one that seems particularly appropriate for synthetic aperture radar (SAR) images is partitioning and shuffling of target in the image (Figure 1). The image of the target is divided in blocks of size $M \times K$, and a random shuffle is operated on the blocks. Notice that by doing so the reflectivity properties of the object are kept, since SAR is associated with the surface reflection. And the shuffled image could still be produced by a metallic object. With the shuffling the location of the point scatters is going to change drastically, creating effectively the signatures of other possible unknown vehicles. In fact, the distance in the grid between the original and the final location of the block should be a parameter in the shuffle. The size of the blocks and the shuffling distance are experimentally determined. We can create many different shuffled images from each training image. This number should also be experimentally determined.

We further propose an evaluation of the shuffled image before it is utilized in the enlarged training set. For that we use the classifier trained without shuffles (called classifier I). Each shuffled image is then presented to the classifier I and not included in the enlarged training set if it produces an output below a threshold. The reasoning is that an image that is too far away from the class discriminant will not affect appreciably its position and should be discarded. All the other cases are kept in the enlarged training set. The value of the threshold is also experimentally determined.

The next aspect is how to include the virtual negative examples for classification. There are more than one possibility: Either the classifier is enlarged with one extra output that will score the shuffles, which implicitly considers the shuffles as an extra class; or the shuffles are associated with the output of all lows, without changing the number of outputs in the classifier. This is more in tune with the concept of negative exemplars, but has the problem of driving to zero the weights of the classifier if the number of shuffles is not appropriately controlled.

It is obvious that many theoretical questions are buried in this methodology, but before pursuing them we would like to experimentally verify the performance of the technique in a realistic setting to judge its promise.

2.2 Classifiers

Two classifiers, one with global discriminant functions, a Perceptron, and the other with local discriminant functions, a Support Vector Machine (SVM), are studied in this paper. When training a pattern classifier, its cost function can be represented in two ways, the empirical risk function and the structural risk function (Vapnik, 1995). The empirical risk is normally calculated with training set error, such as a minimum squared error (MSE) form. The learning criterion of a Perceptron based on MSE is

$$\begin{aligned} \min_w J_1 &= E[y - g(w, x)]^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(y_i - g(w, x_i))^2 \qquad (8) \\ g(w, x) &= \frac{1}{1 + exp(-w^T x)} \qquad (9) \end{aligned}$$

where w is the weight vector of the perceptron. On the other hand, based on the theory of VC-dimension (Vapnik, 1995), the structural risk function consists of two terms, the training set error and confidence interval of the classifier model. With a similar network structure to the perceptron, the learning criterion of a SVM is to minimize the cost function represented as

$$\min_{\alpha} J_2 = \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j - \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) \tag{10}$$

$$g(x) = \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b \tag{11}$$

where $\alpha_i, i = 1, ..., m$, are positive Lagrange multipliers, and $K()$ is a Gaussian function. Those training examples with large values of α_i are called support vectors. Minimizing the first term corresponds to maximizing the margin between classes, while minimizing the second term is to minimize the training set error (in L_1 norm).The classifiers trained both with positive and negative examples are called classifiers II.

3 Experimental Results

Synthetic aperture radar (SAR) ATR experiments are performed in this paper using the MSTAR database to classify targets of interest and reject confusers. The data are 80 by 80 SAR images drawn from three types of ground vehicles: T72, BTR70, and BMP2. The training set contains SAR images taken at a depression angle of 17 degrees, while the testing set is at 15 degrees. The aspect angles are from 0 to 360 degrees. So the SAR images between the training and the testing sets for the same vehicle at the same pose are different, which helps to test the classifier generalization. The sizes of training and testing sets are 698 and 1365, respectively. Also, two types of confusers, 2S1 and D7, are used in the testing.

It's really a hard question to determine exactly the parameters to create the negative examples. We used a small block of 4 × 6 pixels and a random shuffling of the grids within a 5 × 6 rectangle, which can be used as a mask to cover the target contour. We discarded shuffled images with output (from classifier I) below a threshold of 0.5. The total number of shuffles is approximately equal to the size of training set.

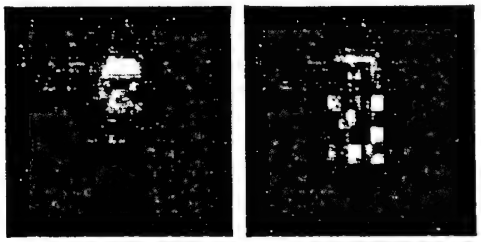


Figure 1: *Left shows a original SAR image; Right gives the shuffled image, which has the same intensity distribution pdf with the original images.*

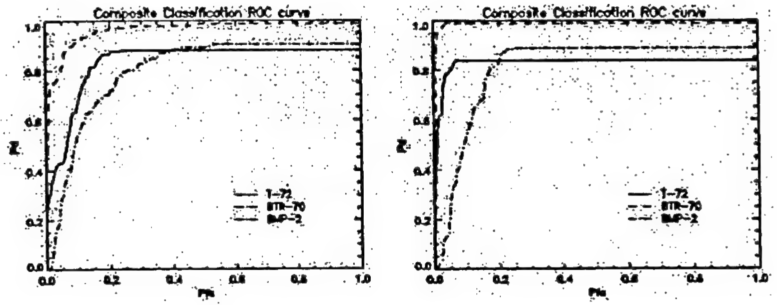


Figure 2: *Left shows the ROC curve of Perceptron-I, which is trained without negative examples; Right gives the ROC curve of Perceptron-II, which is trained with negative examples*

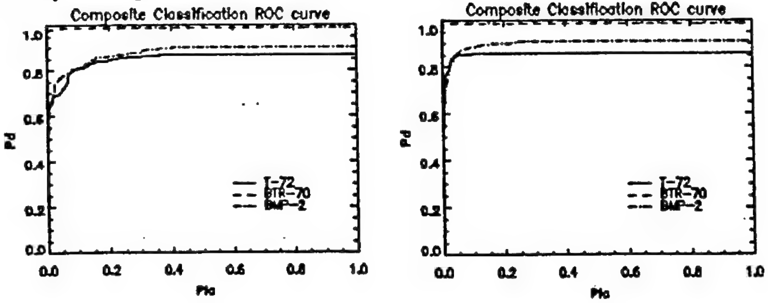


Figure 3: *Left shows the ROC curve of SVM-I, which is trained without negative examples; Right gives the ROC curve of SVM-II, which is trained with negative examples*

Two classifiers are employed, (1) a perceptron trained with the delta rule, with a single layer structure of 6,400 input units and 3 output units; (2) SVM with Gaussian kernels, where the kernel size is chosen as the average Euclidean distance between the training examples. In (Zhao &

Principe, 1999) we have reported results based on the training set. Here we will show our results based on both the training set and the negative examples. To give a general illustration of the performances, Figure 2 and Figure 3 presents the receiver-characteristics-operating (ROC) curves of the classifiers. The classifiers are termed as Perceptron-I and SVM-I when trained without negative examples, and as Perceptron-II and SVM-II when trained with the negative examples. We can immediately observe that the effect of the negative examples is to shift the ROC curves (right panels) towards the left top corner, which means improvement in recognition (fewer false alarms for the same probability of detection). In Figure 2 (the Perceptron) there is a slight decrease in correct classification rates of T-72 when the negative examples are used.

To give a more exact result of the recognition experiment, a threshold is set to keep the probability of target detection P_d in the testing equal to 0.9. Table I show the recognition error rates with respect to the targets, as well as the rejection rates with respect to the confusers.

Table 1: Table I Target Classification Error Rates and Confuser Rejection Rates (in percentage)

Classifier	BMP2	BTR70	T72	Aver.	Rejection
Per-I	9.71	0.0	5.84	6.67	27.19
Per-II	5.28	0.0	12.71	7.69	75.18
SVM-I	4.94	0	7.04	5.12	68.8
SVM-II	5.11	0.51	6.36	4.98	93.25

It is seen that for the Perceptron, although there is a slight decrease of the average target classification performance, the confuser rejection rate increases 48 percents. For the SVM, the average target classification rate gets a little better, and also the confuser rejection rate increases 25 percents. This makes sense, since the perceptron uses a global discriminant so negative exemplars are being used more effectively to make the discriminants closer to the true class. The improvement in a classifier with local discriminant is less dramatic, but it is still substantial.

Both classes of classifiers can present much better confuser rejection performances, which suggests that incorporating negative examples can be used in improving the performance of target recognition systems.

4 Conclusions

This report presents the idea of creating virtual negative examples as severe distortions of the known class patterns. Two classifiers are studied, a perceptron and a Support Vector Machine trained to recognize objects in synthetic aperture radar (SAR) images. They utilize the training set (positive examples) to create the discriminant function of each class in the conventional way. On the other hand, the virtual negative examples will help determine the regions where the discriminant function should yield a low value. The experimental results show that incorporating the negative examples improves greatly (nearly 50 percents improvement) the confuser rejection rates.

Information Theoretic Feature Extraction

1 Introduction

Learning theory develops models from data in an inductive framework. It is therefore no surprise that one of the critical issues of learning is generalization. But before generalizing the machine must learn from the data. How an agent learns from the real world is far from being totally understood. Our most well developed framework to study learning is perhaps statistical learning theory [26], where the goal of the learning machine is to approximate the (unknown) a posteriori probability of the targets given a set of exemplars (Figure 1). But there are many learning conditions that do not fit this model (such as learning without a teacher). Instead we can think that the agent is exposed to sources of information from the external world, and explores and exploits redundancies from one or more sources. This alternate view of learning shifts the problem to the quantification of redundancy and ways to manipulate it. Since redundancy is intrinsically related to the mathematical concept of information, information theory becomes the natural framework to study machine learning. Barlow [2] was one of the pioneers to bring the mathematical concept of information to biologically plausible information processing. His work motivated others to reduce redundancy in learning [10], and it is one of the basis of the work on sparse representations in vision [23]. Linsker proposed the maximization of mutual information between the input to the output of a systems as a principle for self-organization [20].

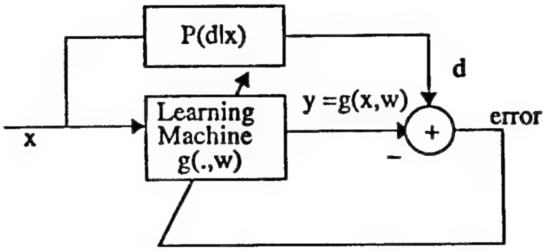


Figure 1: Machine learning according to statistical learning theory. The parameters w are adapted to minimize a measure of the discrepancy between y and d.

Information theory proposed by Claude Shannon [30] has served a crucial role in communication theory [4], but its application to pattern recognition and learning theory has been less pivotal [5]. At the core lies the difficulty that pattern recognition is a discipline based on the learning by example metaphor, while information theory principles require an analytic form for the probability density function (pdf). One possibility is to postulate the form of the pdfs (normally a Gaussian distribution) and estimate from the data their parameters (mean and variance for Gaussian). This has been exactly the way Linsker [20] applied his principle of maximum information preservation (InfoMax). The analytic tractability has also restricted most of the work to linear models [5], [20] [25].

Recently, we have shown that these restrictions are no longer necessary [9], [37]. We developed a nonparametric estimator of entropy for a set of data (based on the Parzen window pdf estimator with appropriate entropy measures) and formulated entropy manipulation as seeking extrema of a cost function. Hence any mapper (linear or nonlinear) can be trained with our scheme. We have shown that the method although computationally demanding ($O(N^2)$, N is the number of data points in the training set) is robust and extracts more information from the input data than the mean square error criterion (which only captures second order information from the data and can be regarded as a specific case of our scheme). We have applied the technique to blind source separation [35] and pose estimation [36] with very good results.

This paper clarifies and extends the algorithm for entropy estimation to the important case of mutual information. The mutual information of two random vectors is a very useful principle for designing information processing systems as InfoMax clearly shows. We will start by briefly reviewing information theoretic learning and its unifying role for learning with or without a teacher. We then proceed by presenting an algorithm that can train arbitrary learning machines to maximize (or minimize) mutual information between its input and output. We will conclude the paper by presenting two applications, one for blind source separation and the other to classification of vehicles in synthetic aperture radar (SAR) imagery.

2 Information Theoretic Learning

We can define information theoretic learning (ITL) as the procedure to adapt the free parameters *w* of a learning machine *g(.,w)* using an information theoretic criterion (Figure 2). Information theoretic learning seems the natural way to train the parameters of a learning machine because the ultimate goal of learning is to transfer the information contained in the external data (input and or desired response) onto the parametric adaptive system. We envisage two basic criteria for ITL: entropy (maximization or minimization) and mutual information (maximization or minimization). Both work in the output space of the learning system, but each has its own domain of application: entropy is a function of one variable and it is not equivariant, i.e. it depends upon the specific coordinate system utilized to represent the data. Hence, entropy manipulation is intrinsically an unsupervised learning paradigm. Entropy maximization is formally an extension of maximizing output energy in linear adaptive systems with the MSE criterion (which leads to the well known principal component analysis), and has been used for blind source separation [3]. Entropy minimization has been utilized for redundancy reduction [2] and can be potentially used in clustering.

Mutual information relies on the estimation of a divergence measure [1] between probability density functions of two random variables and is independent of the coordinate system. Potentially it is the information measure more useful for engineering applications because it involves pairs of random variables. Depending on the nature of these variables mutual information criteria can fall either under supervised or unsupervised learning as we will see below. Mutual information has been utilized in independent component analysis [23], blind source separation [1], and we show

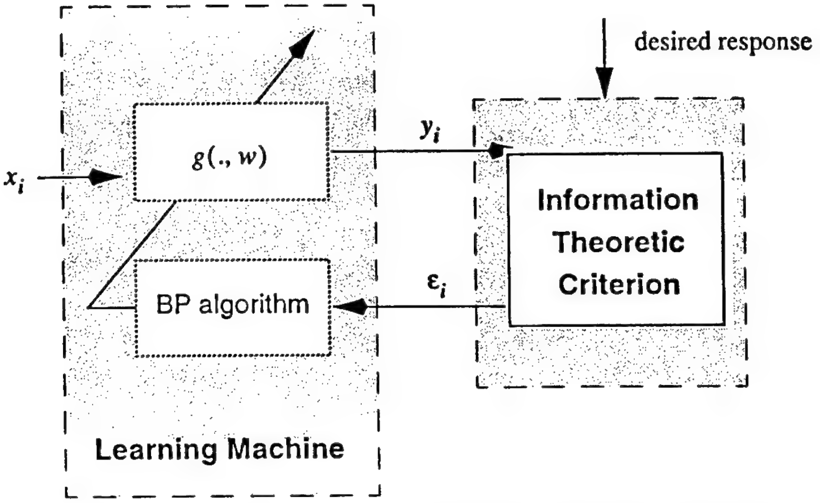


Figure 2: Training a learning machine (linear or nonlinear) with ITL

applications to feature extraction [36], classification [37], and suggest its general role to extend adaptive linear filtering towards information filtering.

2.1 Entropy Criterion and its applications

Let us define the amount of information associated with the measurement of a discrete event x which occurs with probability p as $I(p) = \log\frac{1}{p}$, which is Hartley's measure [18]. Shannon's entropy H_S is the expectation of Hartley's measure, i.e.

$$H_S(x) = \sum_{k=1}^{n} p_k I(p_k) \qquad \sum_{k=1}^{n} p_k = 1 \qquad p_k > 0 \tag{1}$$

Entropy has been extended to continuous random variables $x \in C$ leading to [4]

$$H_S(x) = \int_C p(x)\log\frac{1}{p(x)}dx$$

The zero mean Gaussian probability density function in k dimensional space is

$$G(x,\Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right) \tag{2}$$

where Σ is the covariance matrix. Shannon's entropy for the Gaussian case becomes

$$H(x) = \frac{1}{2}\log|\Sigma| + \frac{k}{2}\log 2\pi + \frac{k}{2} \tag{3}$$

Entropy can lead to an extension of our well-known concept of signal-to-noise ratio (SNR) so useful in engineering. SNR evolved from the need to quantify the deterministic versus the stochastic part o real world signals. SNR is defined as the ratio between the mean and the variance of signal-plus-noise, since normally the signal is deterministic (the mean) and the noise is a wideband (white) zero-mean random variable. If the noise is Gaussian, SNR characterizes adequately the relation between the energy in the mean and in the higher order moments of the measured signal. However, if the noise is not Gaussian, the variance should be replaced by the entropy. More generally, maximization of second order moments plays a central role in learning systems, from adaptive filtering theory [16], to eigendecompositions and neural network learning [15]. Assume that we want to maximize the signal-to-noise (SNR) ratio at the output y of a linear adaptive system with parameters w and multidimensional input x corrupted by noise n, which is assumed white Gaussian with unitary variance. One can show [37] that the solution to this problem leads to the maximization of the Rayleigh quotient

$$J = \frac{|w^T S w|}{|w^T w|} \tag{4}$$

where S is the covariance matrix of the input signal $S = E(xx^T)$. The solution to this problem yields the matched filter for transient signals [15] or the maximum eigenfilter for stationary signals [15] and it is closely related to principal component analysis [6]. Maximization of the Rayleigh quotient is not the only way to maximize output SNR. An alternative, albeit less well-known, criterion for SNR maximization at the output of an adaptive system is [37]

$$J_H = H(w^T x) - H(w^T n) \tag{5}$$

This definition is embedded in Linsker's work on maximum information preservation [20]. Notice that this is a much broader definition of SNR, because now instead of working with the second order statistics of the signal and noise as in Eq. 4 we use their entropies. However, when both the signal x and the noise n are Gaussian distributed, the output entropy due to the signal and due to the noise are both given by Eq. 3, so Eq. 5 becomes

$$J_H = \frac{1}{2}\log\frac{|w^T S w|}{|w^T w|} \tag{6}$$

For optimization Eq. 6 provides the same solution as Eq. 4 due to the monotonic properties of the logarithm. Therefore, for linear systems with Gaussian inputs, maximizing output entropy under the constraint of minimizing output noise entropy defaults to eigen-filtering. As stated by Plumbey [25], the challenge is to develop computational algorithms to extend this formalism to the general case of nonlinear systems and non-Gaussian signals. Bell and Sejnowski used essentially this formulation to solve the cocktail party effect [3].

2.2 Mutual Information Criterion and its applications

Mutual information manipulation is much more useful for learning because it involves the estimation of a distance between pdfs. In fact mutual information between two functions *f(x)* and *g(x)* of the same random variable x can be defined as the Kulback Leibler divergence between the two pdfs [4], i.e.

$$D(f\|g) = \int_C f(x)\log\frac{f(x)}{g(x)}dx \tag{7}$$

The K-L divergence can be regarded as an "asymmetric distance" between the pdfs. One can show that it is always positive and zero only if f(x) = g(x) [4]. For the special case that f(x) is the joint probability of two random variables X_1 and X_2 $f(x) = f_{X_1X_2}(x_1, x_2)$ and g(x) is the product of the corresponding marginal variables $g(x) = f_{X_1}(x_1)f_{X_2}(x_2)$, the Kulback-Leibler divergence becomes the mutual information between X_1 and X_2, that is,

$$I(X_1, X_2) = \iint f_{X_1X_2}(x_1, x_2)\frac{f_{X_1X_2}(x_1, x_2)}{f_{X_1}(x_1)f_{X_2}(x_2)}dx_1dx_2 \qquad (8)$$

Mutual information can also be thought as a distance between the joint density and the product of the marginals since it is always greater or equal to zero. The minimum is obtained when the variables are independent. Mutual information gives rise to either unsupervised or supervised learning rules depending upon how the problem is formulated. Figure 3 shows a block diagram of a unifying scheme for learning based on the same ITL criterion of mutual information. The only difference is the source of information which is shown as a switch with 3 positions.

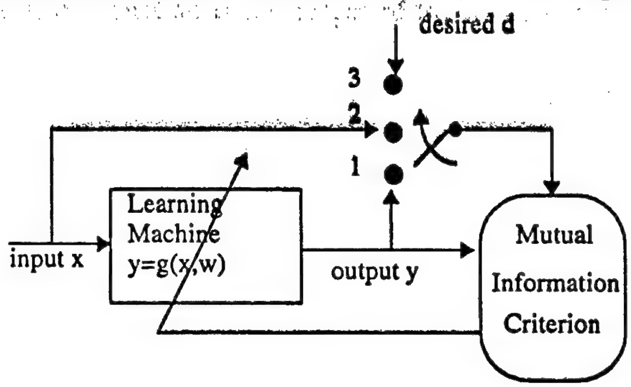


Figure 3: Unifying learning models with the mutual information criterion.

When the switch is in position 1 or 2 learning belongs to the unsupervised type and corresponds to manipulating the mutual information at the output of the learning system or between its input and output. A practical example with switch in position 1 is the on-going work on independent component analysis (ICA) where the goal is to minimize the mutual information among the output of a mapper to yield independent components [15]. An example of the block diagram with switch in position 2 is Linsker's Infomax criterion [20] where the goal is to transfer as much information between the input and output of a mapper by maximizing the joint input-output mutual information.

However, if the goal is to maximize the mutual information between the output of a mapper and an external desired response, then learning becomes supervised. This is achieved by setting the switch to position 3. Note that in this case the desired response appears as one of the marginal pdfs in the mutual information criterion. The two outstanding cases belong both to function approximation: first, if the desired response is a set of indicator functions, the task is classification. However, the desired data is always quantified by means of its pdf, not by deriving a sample by sample error. Therefore we can think of this case as supervised learning without numeric targets, just class labels [37]. Second, if the desired response data is a continuous function then we named the application information filtering [26]. This name came from the realization that the learning machine is seeking a projection of the input space that best approximates (in an information sense) the desired response. In engineering this is the model used for Wiener filtering [16] but where the adaptive system is restricted to be a linear filter and the criterion is minimization of the error variance.

2.3 How appropriate is the mutual information criterion for learning?

One important question that must be answered in the application of mutual information for all these applications is how appropriate it is to fulfill the goals of the processing. Due to the novelty of this approach, we do not have many arguments to justify theoretically the use of mutual information criterion for learning theory. The solid foundation for the use of information theory stems from communication theory [30], [4], [8], and from statistical mechanics [17]. But in learning theory the fundamental problem is inference and statistical estimation [32]. For instance in parameter estimation, we know today that any unbiased estimator is bounded from below by the Cramer-

Rao bound [5]. It is important to ask a similar question for mutual information based criteria. We only know of a result stated by Fano for the case of classification [8]. Assume that the goal is to estimate a variable x with a discrete pdf $p(x)$ by calculating an estimate $\hat{x}$ from another random variable y characterized by $p(x|y)$. Under mild conditions

$$P(x \neq \hat{x}) \geq \frac{H_S(x|y) - 1}{\log(\Theta(x))} \tag{9}$$

where Θ(x) means the number of possible instances of x. This equation shows that the probability of error is lower bounded by the conditional entropy of x given y. Substituting the definition of mutual information $I(x, y)$ we obtain

$$P(x \neq \hat{x}) \geq \frac{H_S(x) - I(x, y) - 1}{\log(\Theta(x))} \tag{10}$$

Notice that we have no control over the entropy of x nor the number of possible instances of x. *Therefore to improve the lower bound on the achievable probability of error, we should maximize the mutual information between x and y.* We may think that Eq. 10 is not a very useful result because it does not provide an upper bound for the probability of error. But exactly like the Cramer-Rao bound, it can not because an estimator for mutual information is not specified in the formulation. Eq. 10 talks about the achievable lower bound, while the upper bound depends upon the particular estimator we choose. There are very interesting relationships between the Fisher information matrix and mutual information, but we can not elaborate them here.

With all these nice properties of information measures, the reader may be wondering why information theory has not been widely applied in machine learning. The answer lies in the difficulty of estimating entropy and mutual information directly from data. Next we will provide an estimator for entropy based on an alternative definition of entropy proposed by the Hungarian mathematician Alfred Renyi [28].

3 Renyi's entropy

Shannon's entropy was defined in Eq. 1 as the expectation of Hartley's amount of information, but there are alternate definitions of information. In the general theory of means, the mean of the real numbers $x_1, ..., x_n$ with weights $p_1, ..., p_n$ has the form:

$$\bar{x} = \varphi^{-1}\left(\sum_{k=1}^{n} p_k \varphi(x_k)\right) \tag{11}$$

where $\varphi(x)$ is a Kolmogorov-Nagumo function, which is an arbitrary continuous and strictly monotonic function defined on the real numbers. In general, an entropy measure obeys the relation:

$$H = \varphi^{-1}\left(\sum_{k=1}^{n} p_k \varphi(I(p_k))\right) \tag{12}$$

As an information measure, $\varphi(\ )$ can not be arbitrary since information is "additive". To meet the additivity condition, $\varphi(\ )$ can be either $\varphi(x) = x$ or $\varphi(x) = 2^{(1-\alpha)x}$. If the former is used, Eq. 12 will become Shannon's entropy. If $\varphi(x) = 2^{(1-\alpha)x}$, Eq. 12 becomes Renyi's entropy with order α [28] which we will denote by $H_{R\alpha}$

$$H_{R\alpha} = \frac{1}{1-\alpha}\log\left(\sum_{k=1}^{n} p_k^{\alpha}\right) \qquad \alpha > 0, \alpha \neq 1 \tag{13}$$

When $\alpha = 2$, Eq. 13 becomes $H_{R2} = -\log\sum_{k=1}^{n} p_k^2$ and it will be called Quadratic Entropy.

For the continuous random variable Y with pdf $f_Y(y)$, we can obtain the differential version for these two types of entropy following a similar route to the Shannon differential entropy [30]:

$$\begin{cases} H_{R\alpha}(Y) = \dfrac{1}{1-\alpha}\log\left(\displaystyle\int_{-\infty}^{+\infty} f_Y(y)^{\alpha} dy\right) \\ H_{R2}(Y) = -\log\left(\displaystyle\int_{-\infty}^{+\infty} f_Y(y)^{2} dy\right) \end{cases} \tag{14}$$

From the point of view of estimation, Renyi's entropy is very appealing since it involves the integral of a power of the pdf, which is much simpler to estimate than the pdf. Renyi's entropy also brings a different view to the problem of entropy estimation. Let us consider the probability distribution $P= (p_1,p_2,...,p_N)$ as a point in a N-dimensional space. Due to the conditions on the probability measure ($p_k \geq 0, \sum_{k=1}^{N} p_k = 1$) P always lies in the first quadrant of an hyperplane in N dimensions intersecting each coordinate axis at the point 1 (Fig. 4). The distance of P to the origin is

$$\|P\|_{\alpha} = \sqrt[\alpha]{\sum_{k=1}^{N} p_k^{\alpha}} = \sqrt[\alpha]{V_{\alpha}} \tag{15}$$

and is called the α-norm of the probability distribution [13]. Renyi's entropy (Eq. 13) can be written as a function of V_{α}

$$H_{R\alpha} = \frac{1}{1-\alpha}\log V_{\alpha} \tag{16}$$

When different values of α are selected in the Renyi's family, the end result is to select different α-norms. Shannon entropy can be considered as the limiting case $\alpha \to 1$ of the probability distribution norm. Other values of α will measure the distance to the origin in different ways, very much like the selection of the norm of the error in the learning criterion [15]. We settled on $\alpha = 2$

because in the nonlinear dynamics literature Renyi's entropy has also been used to estimate dimension from experimental data with very good results [12]. In general, higher α increases the robustness of the estimation in areas with low sample density, but the algorithmic complexity increases exponentially with α, so $\alpha = 2$ is a good compromise.

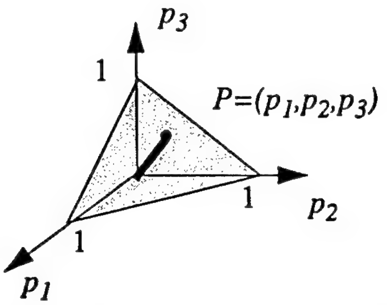


Figure 4: Geometric interpretation of entropy for N=3. The distance of P to the origin is related to the α-norm.

It is important to discuss the implications of this development. Shannon's entropy definition has been intrinsically related to the estimation of the pdf of the random variable. All attempts of using it have either assumed an analytical model for the pdf [4], [18] or have used nonparametric pdf estimators [34], which perform poorly in large dimensionality spaces. Renyi's definition alternatively shows that entropy is related to the norm of the pdf in probability spaces. The norm of a vector is a much easier quantity to estimate in high dimensional spaces, in particular if the order of the norm is low (such as the 2-norm).

4 Quadratic entropy and its nonparametric estimator

We will be working with Renyi's quadratic entropy because we have found a way of estimating the 2-norm of the pdf using the well known Parzen window estimator [24]. Let $y_i \in R^k, i = 1, \ldots, N$, be a set of samples from a random variable $Y \in R^k$ in k-dimensional space which can be the output of a nonlinear mapper such as a multilayer perceptron (MLP). How can we estimate the 2-norm of this set of data samples? One answer lies in the estimation of the data pdf by the Parzen window method using a Gaussian kernel:

$$f_Y(y) = \frac{1}{N}\sum_{i=1}^{N} G(y - a_i, \sigma^2) \tag{17}$$

where $G(y, \Sigma)$ is the Gaussian kernel in k dimensional space, and Σ is the covariance matrix (here spherically symmetric kernels $\Sigma = \sigma^2 I$ will be utilized). We just need to substitute Eq. 17 in Eq. 14 to yield immediately:

$$\begin{aligned} H(\{y_i\}) &= -\log\left(\int_{-\infty}^{+\infty} f_Y(y)^2 dy\right) = -\log V(\{y_i\}) \\ V(\{y_i\}) &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(y_i - y_j, 2\sigma^2) \end{aligned} \tag{18}$$

Making the analogy between data samples and "physical particles", $V(\{y_i\})$ can be regarded as an overall potential energy of the data set since $G(y_i - y_j, 2\sigma^2)$ can be taken as the potential energy of data sample y_i in the potential field of data sample y_j, or vice versa. *We will call this potential energy an information potential, where the data samples have a correspondence to physical particles and the information potential to a potential field.* So, maximizing Renyi's quadratic entropy in this case is equivalent to minimizing information potential. Our estimator for quadratic Renyi's entropy (Eq. 18) only suffers from the approximation inherent to the pdf estimation.

Just like in mechanics, the derivative of the potential energy is a force, in this case an *information force*. The information force moves the data samples in the output space to achieve an equilibrium state dictated by our criterion. Therefore,

$$\frac{\partial}{\partial a_i}G(y_i-y_j, 2\sigma^2) = G(y_i-y_j, 2\sigma^2)(y_j-y_i)/(2\sigma^2) \tag{19}$$

can be regarded as the force that the data sample y_j impinges upon y_i. If all the data samples are free to move in a certain region of the space, then the information forces between each pair of data $\partial G(y_i-y_j, 2\sigma^2)/\partial y_i$ or $\partial G(y_i-y_j, 2\sigma^2)/\partial y_j$ will drive all the data samples to a state with minimum information potential.

Suppose the data samples are the outputs of our parametric adaptive system, for example an MLP. If we want to adapt the MLP such that the mapping maximizes the entropy at the output $H(\{y(n)\})$, the problem is equivalent to finding the parameters of the MLP so that the information potential $V(\{y(n)\})$ is minimized. Therefore, the information forces applied to each data sample can be back-propagated to the parameters using the chain rule [29]. As an example, the following gradient can be interpreted as force back-propagation:

$$\frac{\partial}{\partial w_{ij}}V(\{y(n)\}) = \sum_{n=1}^{N}\sum_{p=1}^{k}\frac{\partial}{\partial y_p(n)}V(\{y(n)\})\frac{\partial}{\partial w_{ij}}y_p(n) \tag{20}$$

where $y(n) = (y_1(n), \dots, y_k(n))^T$, and w_{ij} is one of the weights in MLP. The quantity

$$\frac{\partial}{\partial y(n)}V(\{y(n)\}) = \left(\frac{\partial}{\partial y_1(n)}V(\{y(n)\}), \dots, \frac{\partial}{\partial y_k(n)}V(\{y(n)\})\right)^T \tag{21}$$

is the information force that the data sample $y(n)$ is subject to. Notice that the sensitivity of the output with respect to a MLP parameter $\frac{\partial}{\partial w_{ij}}y_p(n)$ is the transmission mechanism through which information forces are back-propagated to the parameter. From the analogy with the backpropagation formalism we conclude that information forces take the place of the injected error. *So, we obtain a general, nonparametric, and sample-based methodology to adapt arbitrary nonlinear (smooth) mappings for entropy manipulation.*

The user has to select only two parameters in the training algorithm: the learning rate and the kernel size. The learning rate is annealed during learning with a linear rule. From the understanding of the information potential, it is straight forward to conclude that the samples have to interact with each other. Therefore for entropy minimization we set the kernel size such that the two furthest samples still interact. Since the samples change position during learning, this distance should be updated during training (but infrequently to avoid adding another dynamics to the learning process). For entropy maximization the goal is to produce heavy interaction in the beginning of training (same rule as before) and slowly anneal the kernel size, as done in Kohonen training [19]. We verified experimentally that the kernel size needs to be in the correct range, but does not need to be finely tuned. In [27] we present a more principled approach to set the kernel size based on cross-validation.

5 Quadratic Mutual Information and Cross-Information Potential

For two random variables Y_1 and Y_2 (with marginal pdfs $f_{Y_1}(y_1)$, $f_{Y_2}(y_2)$ and joint pdf $f_{Y_1Y_2}(y_1, y_2)$), mutual information can be estimated using the Kulback-Leibler divergence between the joint probability and the factored marginals [5]. Inspired by this result and constrained by using quadratic forms of pdfs, we propose the following distance based on the Cauchy-Schwartz inequality:

$$I_{CS}(Y_1, Y_2) = \log\frac{\left(\int\int f_{Y_1Y_2}(y_1, y_2)^2 dy_1\, dy_2\right)\left(\int\int f_{Y_1}(y_1)^2 f_{Y_2}(y_2)^2 dy_1 dy_2\right)}{\left(\int\int f_{Y_1Y_2}(y_1, y_2) f_{Y_1}(y_1) f_{Y_2}(y_2) dy_1 dy_2\right)^2} \tag{22}$$

which we called Cauchy-Schwartz quadratic mutual information (CS-QMI) [27]. It is obvious that $I_{CS}(Y_1, Y_2) \geq 0$ and the equality holds true if and only if Y_1 and Y_2 are statistically independent, i.e. $f_{Y_1 Y_2}(y_1, y_2) = f_{Y_1}(y_1)f_{Y_2}(y_2)$. So, $I_{CS}(Y_1, Y_2)$ is an appropriate measure for the independence of two variables (minimization of mutual information). We also have experimental evidence that $I_{CS}(Y_1, Y_2)$ is an appropriate measure for dependence of two variables (maximization of mutual information). Although we are unable to provide yet a strict justification that $I_{CS}(Y_1, Y_2)$ is appropriate to measure dependence, we still call Eq. 22 "Quadratic Mutual Information" because of both convenience and the fact that it only relies on the quadratic forms of pdfs. In [27] we proposed also an alternative definition for quadratic mutual information based on the Euclidean distance between the joint entropy and the product of their marginals, i.e.

$$I_{ED}(Y_1, Y_2) = \iint (f_{Y_1 Y_2}(y_1, y_2) - (f_{Y_1}(y_1))f_{Y_2}(y_2))^2 dy_1 dy_2 \qquad (23)$$

which was named Euclidean distance quadratic mutual information (ED-QMI). Unlike the estimator of Eq. 18 for Renyi's entropy, Eq. 22 and 23 are being investigated in our laboratory as proxies (approximations) of the mutual information criterion.

For learning, what is essential is that the minima and maxima of the newly defined CS-QMI and ED-QMI coincide with the extrema of I(Y_1, Y_2). We have derived the relationships between CS-QMI, ED-QMI and mutual information for the case of Gaussian random variables [37], and concluded that they have the same maxima and minima. In [27] we show a case of a simple probability mass function to illustrate that the extrema between CS-QMI, ED-QMI and mutual information also coincide. For more general pdfs we only have experimental evidence that the quadratic mutual information criteria are able to find solutions that produce good results. Here we will present the derivation for I_{CS} (see [27] for a full treatment).

Suppose that we observe a set of data samples $\{y_{i1}, i = 1, \ldots, N\}$ for the variable Y_1, $\{y_{i2}, i = 1, \ldots, N\}$ for the variable Y_2. Let $y_i = (y_{i1}, y_{i2})^T$. Then $\{y_i, i = 1, \ldots, N\}$ are data samples for the joint variable $(Y_1, Y_2)^T$. Based on the Parzen window method, the joint pdf and marginal pdf can be estimated as:

$$\begin{cases} f_{Y_1Y_2}(y_1, y_2) = \frac{1}{N}\sum_{i=1}^{N} G(y_1 - y_{i1}, \sigma^2)G(y_2 - y_{i2}, \sigma^2) \\ f_{Y_1}(y_1) = \frac{1}{N}\sum_{i=1}^{N} G(y_1 - y_{i1}, \sigma^2) \\ f_{Y_2}(y_2) = \frac{1}{N}\sum_{i=1}^{N} G(y_2 - y_{i2}, \sigma^2) \end{cases} \qquad (24)$$

Combining (22), (24) and using (18), we obtain the following expressions for the CS-QMI based on a set of data samples:

$$\begin{cases} I_{CS}((Y_1, Y_2)|\{y_i\}) = \log\frac{V(\{y_i\})V_1(\{y_{i1}\})V_2(\{y_{i2}\})}{V_{nc}(\{y_i\})^2} \\ V(\{y_i\}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(y_i - y_j, 2\sigma^2) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\prod_{l=1}^{2} G(y_{il} - y_{jl}, 2\sigma^2)\right) \\ V_l(y_j, \{y_{il}\}) = \frac{1}{N}\sum_{i=1}^{N} G(y_{jl} - y_{il}, 2\sigma^2), \quad l = 1, 2 \\ V_l(\{y_{il}\}) = \frac{1}{N}\sum_{j=1}^{N} V_l(y_j, \{y_{il}\}), \quad l = 1, 2 \\ V_{nc}(\{y_i\}) = \frac{1}{N}\sum_{j=1}^{N}\left(\prod_{l=1}^{2} V_l(y_j, \{y_{il}\})\right) \end{cases} \qquad (25)$$

These expressions can be interpreted in terms of information potentials and extended for the case of multiple variables [37], but we do not have space to elaborate on the interpretation.

The cross-information potential (the argument of the log of I_{CS} in Eq. 25) is more complex than the information potential of Eq. 18. Three different potentials (joint potential *V(.)*, marginal potentials $V_l(.)$, unnormalized cross-potential $V_{nc}(.)$) contribute to the cross-information potential. Hence, the force applied to each data sample y_p comes from three independent sources (the marginal components). The q marginal force (marginal space indexed by q) that the data point y_p receives can be calculated according to the following formulas:

$$
\begin{aligned}
&\frac{\partial}{\partial y_{pq}} V(\{y_i\}) = \frac{1}{N^2} \sum_{i=1}^{N} \left(\prod_{l=1}^{k} G(y_{iq} - y_{pq}, 2\sigma^2) \right) \frac{y_{iq} - y_{pq}}{\sigma^2} \\
&\frac{\partial}{\partial y_{pq}} V_q(\{y_{iq}\}) = \frac{1}{N^2} \sum_{i=1}^{N} G(y_{iq} - y_{pq}, 2\sigma^2) \frac{y_{iq} - y_{pq}}{\sigma^2} \\
&\frac{\partial}{\partial y_{pq}} V_{nc}(\{y_i\}) = \frac{1}{N^2} \sum_{j=1}^{N} \frac{1}{2}(B_j) G(y_{jq} - y_{pq}, 2\sigma^2) \frac{y_{jq} - y_{pq}}{\sigma^2}
\end{aligned}
\qquad (26)
$$

where $B_j = \prod_{l \neq q} V_l(y_j, \{y_{il}\}) + \prod_{l \neq q} V_l(y_p, \{y_{il}\})$. The overall marginal force that the data point y_p receives is:

$$
\begin{aligned}
&\frac{\partial}{\partial y_{pq}} I_{CS}((Y_1, Y_2) | \{y_i\}) = \\
&= \frac{1}{V(\{y_i\})} \frac{\partial}{\partial y_{pq}} V(\{y_i\}) + \frac{1}{V_q(\{y_{iq}\})} \frac{\partial}{\partial y_{pq}} V_q(\{y_{iq}\}) - 2 \frac{1}{V_{nc}(\{y_i\})} \frac{\partial}{\partial y_{pq}} V_{nc}(\{y_i\})
\end{aligned}
$$

Notice that the force from different sources are normalized by their corresponding information potentials to balance them out. This is a very nice feature (equivariant) of the CS-QMI. Once the forces that each data point receives are calculated, these forces function as the injected error, and can again be back-propagated to all the parameters of the learning machine so that the adaptation takes the system state to the extremum of the criterion (minimum or maximum depending on the sign of the error).

6 Experimental results

In order to demonstrate the use of ITL in realistic problems, we will present here an example of blind source separation and classification. Other tests of this methodology have been reported [35],[31], [36].

6.1 Blind source separation

Blind source separation can be formulated in the following way. The observed data $X = AS$ is a linear mixture ($A \in R^{m \times m}$ is non-singular) of independent source signals $S = (S_1, \ldots, S_m)^T$. There is no further information about the sources and the mixing matrix, hence the denomination "Blind". The problem is to find a projection $W \in R^{m \times m}$, so that $Y = WX$ will become $Y = S$ up to a permutation and scaling.

We present below the results of a linear de-mixing system trained with the Cauchy-Schwartz quadratic mutual information (CS-QMI) criterion. From this point of view, the problem can be restated as finding a projection $W \in R^{m \times m}$, $Y = WX$ so that the CS-QMI among all the components of Y is minimized, that is all the output signals are independent of each other. For ease of illustration, only 2-source-2-sensor problem is tested.

There are two experiments presented: Experiment 1 tests the performance of the method on a very sparse data set which was instantaneously mixed in the computer with a mixing matrix [2, 0.5; 1, 0.6]. Two, 2-D, different colored Gaussian noise segments are used as sources, with 30 data points for each segment (sparse data case). The two segments were concatenated and shuffled. Fig. 5 (left panel) shows the source density in the joint space (each axis is one source signal). As Fig. 5 shows, the mixing produces a mixture with both long and short "tails" which is difficult to separate (middle panel). Whitening is first performed on the mixtures to facilitate de-mixing. The data distribution for the recovered signals are plotted in Fig. 5 (right panel). As we can observe the original source density is obtained with high fidelity.

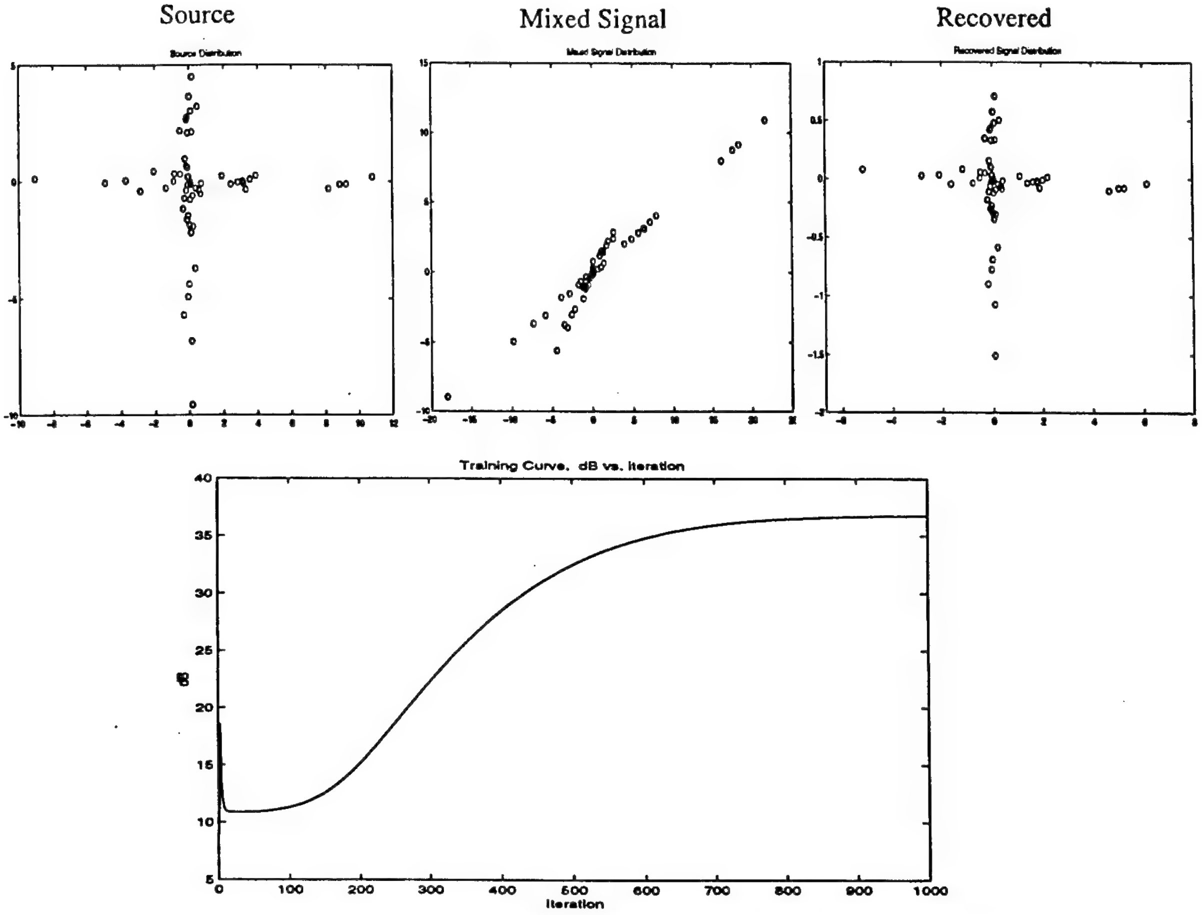


Figure 5: Data distributions for the sources (left), mixed (middle) and demixed with the proposed method (right). Learning curve on bottom plotting the product WA in dB as a function of batch iterations. Notice the final 36 dB of SNR.

Fig. 5 also contains the evolution of the SNR of de-mixing-mixing product matrix (WA) during training as a function of batch iterations. The adaptation approaches a final SNR of 36.73 dB in less than 700 batch iterations.

Experiment 2 uses two speech signals from the TIMIT database as source signals (Fig. 6). The mixing matrix is [1, 3.5; 0.8, 2.6] where the two mixing direction [1, 3.5] and [0.8, 2.6] are similar. An on-line implementation is tried in this experiment, in which a short-time window (200 samples) slides over the speech data (e.g. 10 samples/step). In each window position, the speech data within the window is used to calculate the information potentials, information forces and back-propagated forces all using batch learning to adjust the de-mixing matrix. As the window slides at 10 samples/step the demixing matrix keeps being updated. The training curve (SNR vs. sliding index) is shown in Fig. 6 which tells us that the method converges within 40,000 samples of speech and achieves a SNR approaching 49.15 dB, which is comparable to other methods for this mixing condition. The large spikes in the training curve shows the occasional almost perfect demixing matrix estimation while the algorithm is still adapting (notice that during adaptation the algorithm can estimate one of the directions very well although it is still far away from the optimal solution). In order to obtain a stable result the learning rate is linearly reduced through training. Although whitening is done before CS-QMI learning, we believe that the whitening process can also be incorporated into the ITL algorithm.

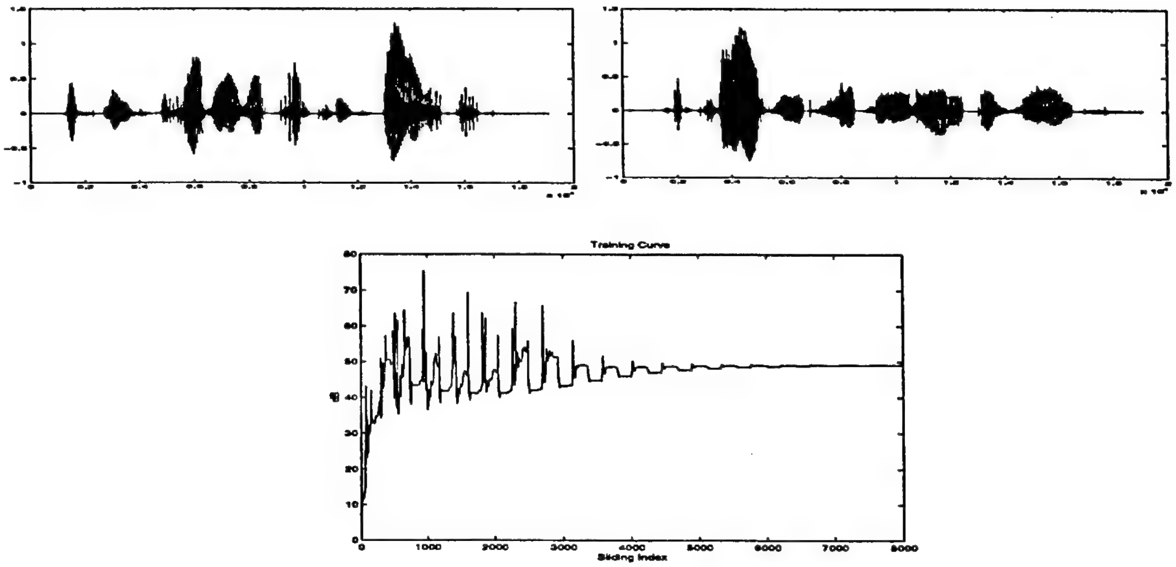


Figure 6: Two speech signals from TIMIT that were mixed, and resulting training curve plotting WA in dB versus the sliding window index. Final SNR is around 50 dB.

6.1.1 Blind source separation of High Range Resolution (HRR) profiles.

We also studied the application of this algorithm to the separation of signatures from closely separated targets. Let us assume that we have two vehicles parked side by side at small proximity of each other. Close proximity is here measured in relation to the radar cross section. When the radar beam illuminates the targets there will be a mixing of the two signatures. We assume that this mixing is going to be instantaneous, i.e. that each reflection is going to be multiplied by an unknown scalar before being added together. Hence the overall signature follows the model of blind source separation if we assume that the each vehicle signature is statistically independent of the other, which seems a reasonable assumption.

We created a data set built from HRRs of two different vehicles (T72 and BMP2) taken independently at 10 degrees elevation and zero degrees azimuth (Xpatch data). These two signatures were mixed with two different scalars to create a single look. The process was repeated with a different set of constants to create a second look. These scalars comprise the mixing matrix M (see figure 7), and are unknown to the algorithm. We will use the minimization of mutual information to try to demix the two original HRR profiles from the two looks. Notice that we have to have as many measurements (looks) as there are targets. Figure 7 shows the original signatures and the mixed signatures that are the output to our 2 input-2 output mixing system, which was a two input two output perceptron with tanh nonlinearity. The algorithm was run until convergence, and the results are shown in the right panel of the following Figure

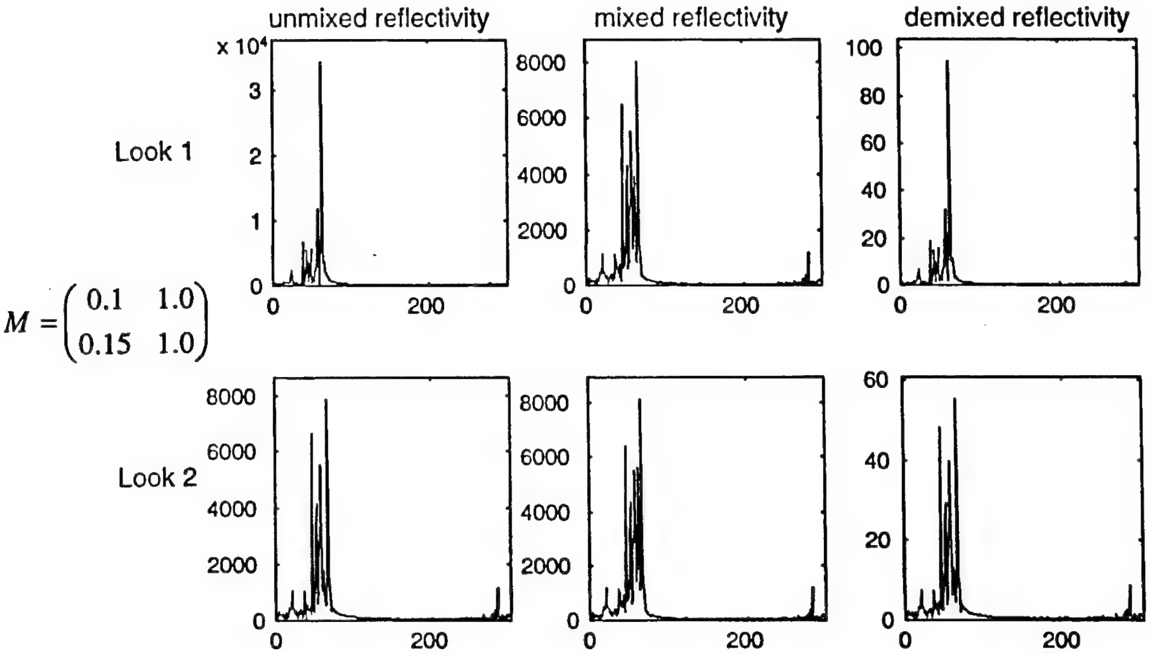


Figure 7: Downrange profiles of two targets, the mixed profiles and the mixing matrix, and the results of the demixing algorithm.

These results are very encouraging since they show that the mixed signatures (middle panels) can be un-mixed by minimizing the mutual information among the outputs of the demixing system using the ITL algorithm. These results were repeatable. Figure 8 shows another case with a different mixing matrix and the results are basically the same.

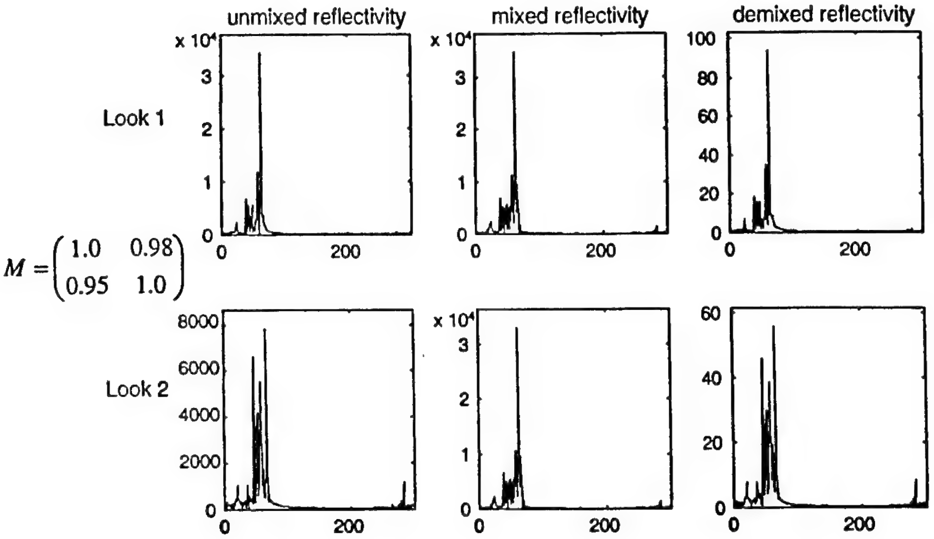


Figure 8: A different artificially mixed profiles. The algorithm is able to demix the two signatures.

With these positive results, we then went to a more realistic testing. We obtained from Veridian a set of XPATCH data that simulated the signatures of two T72 tanks over a flat conducting plane and parked side by side (separated by 12 inches) as shown in Figure 9.

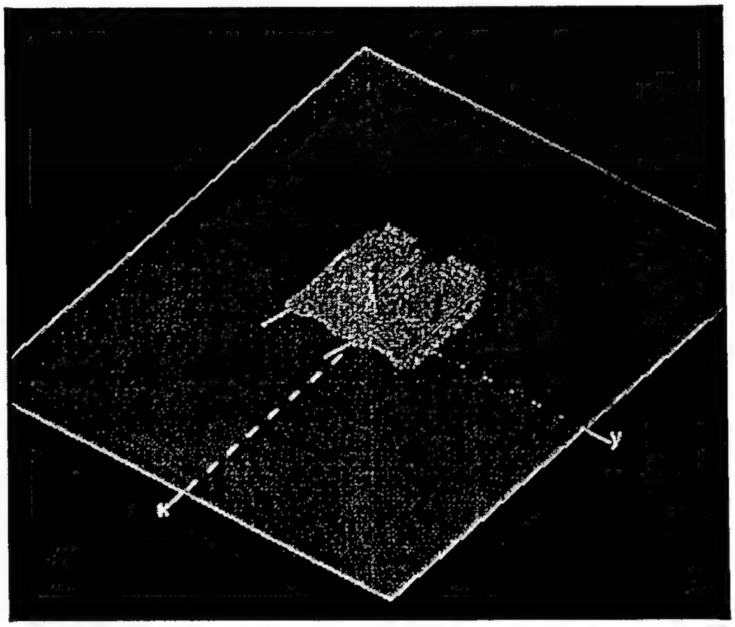


Figure 9: Illustration of the Xpatch simulation. Two T72s were parked at close proximity over a conducting ground plane.

The azimuth angles are 350 (minus 10), 352 (minus 8), 354 (minus 6), and 356 (minus 4) and the elevation is 15 degrees. The resolution is 1.681 inches with a 40 dB Kaiser weighting. Figure 10 shows each range profile. Notice that the two signatures are very similar due to the fact that the vehicles are the same, and their spacing is very small compared with the relative angle. Hence this is a very difficult problem.

We applied the algorithms based on minimization of mutual information that successfully separated speech and the synthetically mixed range profiles, but the results were rather poor. There was no noticeable change in the profiles, which mean that the algorithm did not converge to the right solution. We could not obtain from Veridian in time the profile of just one tank, i.e. we did not have ground truth available to us.

We believe now that this problem can not be separated by an instantaneous mixing system, because the model for the generation of the data changes with range, i.e. it is a convolutive model. This may explain the unsuccessful demixing. So our conclusion is that much more work needs to be put into this part of the project.

7 Classification

This example is part of an on-going effort in our laboratory to develop classifiers for automatic target recognition using synthetic aperture radar (SAR/ATR) imagery. Synthetic aperture radar (SAR) automatic target recognition (ATR) experiments were performed using the MSTAR database to classify three targets and reject confusers. The data are 80 by 80 SAR images drawn from three types of ground vehicles: the T72, BTR70, and BMP2 as shown in Figure 10(c). These images are a subset of the 9/95 MSTAR Public Release Data [21]. The poses (aspect angles) of the vehicles lie between 0 to 180 degrees as shown in Figure 10(b).

A SAR image is the amplitude of the FFT (fast Fourier transform) of the radar return properly mapped from time to space. The images are very noisy due to the image formation and lack resolution due to the radar wavelength, which makes the classification of SAR vehicles a non-trivial problem [33]. Unlike optical images, the SAR images of the same target taken at different aspect angles are not correlated with each other which precludes the existence of a rotation invariant transform. This results from the fact that a SAR image reflects the fine target structure (point scatter distribution on the target surface) at a certain pose. Parts of the target structure will be occluded when illuminated by the radar, which results in dramatic differences from image to image with angular increments as small as 10 degrees. Thus a classifier has to be trained with each pose.

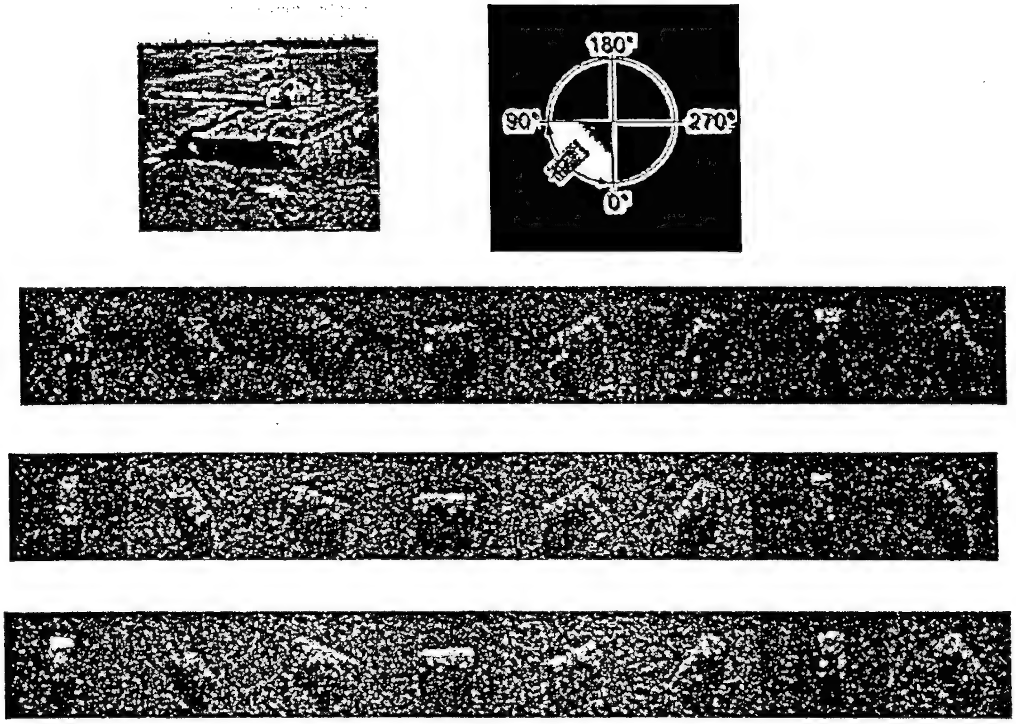


Figure 10: Examples of the SAR training set. Notice the difficulty of the task both in terms of the variability and noise in the images.

In these experiments we have created 6 classifiers each covering 30 degrees of aspect such that vehicles appearing at poses between 0-180 degrees can be classified accurately. We have further compared three classifiers: a support vector machine (SVM) using a Gaussian kernel [32], an optimal separation hyperplane (OH) classifier [38] and the classifier based on the mutual information criterion ED-QMI of Eq. 23. We compare them with the perceptron to gauge the level of performance with more conventional methods.

The training set contained SAR images taken at a depression angle of seventeen degrees, while the testing set depression angle is fifteen degrees. Hence, the SAR images between the training and the testing sets for the same vehicle at the same pose are different, which helps to test the classifier generalization. Variants (different serial number) of the three targets were also used in the testing set. The size of training and testing sets is 406 and 724, respectively.

Two types of experiments were conducted. One is the conventional classification task, and the other is the more challenging recognition task. In the recognition task confuser vehicles, i.e. other vehicles not used in the training where presented to the classifiers and the rejection rate was computed for a detection probability of Pd=0.9.

The SVM and OH classifiers where trained with the Adatron algorithm [11]. The difference between these two classifiers is that the OH does the classification in the input space, while the SVM does the classification in feature space. For this problem nearly all the inputs are support vectors so the classification with the SVM is in fact done in a 400 dimensional space. Further details can be found in [38].

The classifier based on the ED-QMI is a linear classifier with a 80x80 input layer and 3 outputs. Due to the large input dimension, an one hidden layer MLP produced virtually the same results. The idea is to find a projection that will preserve the most information jointly contained in the output and the desired response. Therefore, one should maximize our measure of mutual information in the criterion (ED-QMI). The training progresses smoothly and is over in 200 batch iterations. Figure 11 depicts three snapshots of training in the beginning of training, half way and at the end of training. In the left panels we show the samples and the information forces being exerted on each output sample. In the right panels we zoom in the output space to have a clearer view of the separation between clusters. Notice that in the beginning of training the images of each input are mixed in the output space indicating bad discrimination. Half way through the training we see the clusters separating, and the information forces are large and centrifugal, i.e. separating the cluster images. We can also observe a smaller dispersion in each cluster because information forces among samples of different clusters repel while the samples of each class attract. At the end of training the information forces are almost zero and the clusters are well separated, and very compact (almost a point). Clearly this will provide easy discrimination among the classes (at least for the training set). Note that the ED-QMI information force in this particular case can be interpreted as repulsion among the samples with different class labels, and attraction with each other among the samples within the same class.

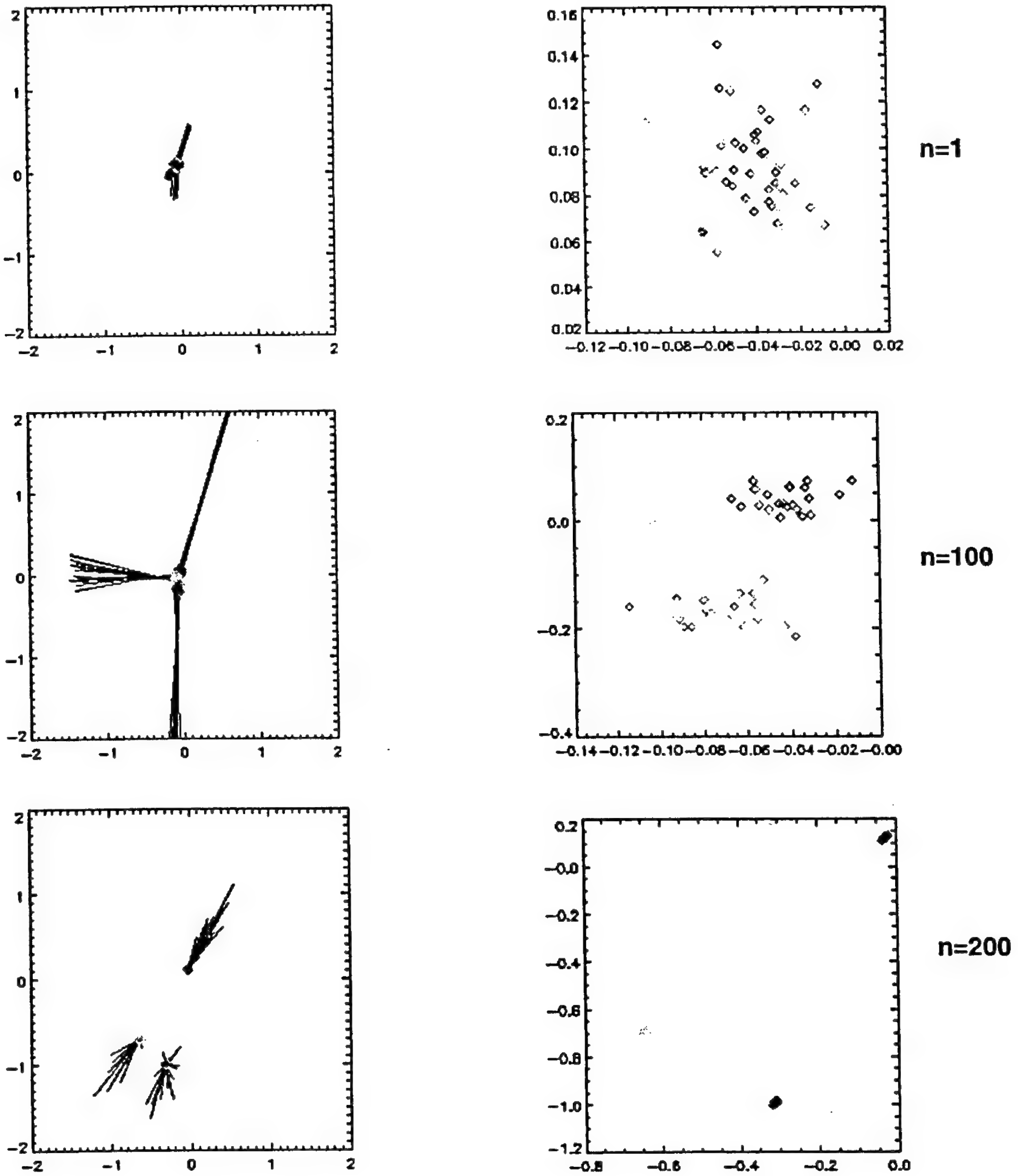


Figure 11: Three snapshots of the 2D output space of the classifier during learning. Left panels show the output samples (color coded per class) and the information forces, while the right panels are zooms of the output space to see each individual output samples.

The joint pdf of the class labels and the mapper output $f_{CY}(y, c)$ is the natural "by-product" of our training scheme. Actually, the cross information potential for classification is based on the Parzen window estimation of the joint pdf

$$f_{CY}(y, c) = \frac{1}{N}\sum_{i=1}^{N} G(y - y_i, 2\sigma^2)\delta(c - c_i) \tag{27}$$

where σ^2 is the variance for Gaussian kernel function for the feature variable y, $\delta(c - c_i)$ is the Kronecker delta function; i.e.,

$$\delta(c - c_i) = \begin{cases} 1 & c = c_i \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Based on the joint pdf $f_{CY}(y, c)$, the Bayes classifier can be built up as

$$\hat{c} = \arg\underset{c}{max}\, f_{CY}(y, c) \qquad y = g(x, w) \tag{29}$$

Since the class identity variable C is discrete, the search for the maximum can be simply implemented by comparing each value of $f_{CY}(y, c)$.

Table I shows the results for classification using the OH, the SVM, the ED-QMI and the perceptron.

Table I. Classification error (%)

	BMP2	BTR70	T72	Average
OH	6.45	1.87	5.28	5.25
SVM	7.74	0.93	4.56	5.39
ED-QMI	6.77	2.80	4.23	5.11
Perceptron	9.35	0.93	11.4	8.98

We see that the classifier trained with ED-QMI performs at the same level as the other two classifiers. This is very rewarding since the SVMs are known for their extremely good performance. It is interesting to analyze the principles behind each classifier. The OH is creating discriminant functions in the input space (6,400 dimensions), while the SVM is creating discriminant functions in a space of dimensionality given by the support vectors. This decoupling between the input space and the feature space dimensionality is a distinct feature of the SVMs. In our case this yields a smaller 400 dimensional space. The ED-QMI is also creating discriminant functions in the input space, but it is using a mutual information criterion to choose the projections. From Table I we see that the ED-QMI result is slightly better (although the differences may not be significant), which means that the ED-QMI is positioning the discriminant functions for small classification errors using the structure of the data clusters. As a comparison to the "conventional" classifiers we also implemented a perceptron and trained it with weight decay and early stopping (for details see [15]). As we can observe from Table I the perceptron works with almost twice the misclassification error, even after all the care of controlling the number of degrees of freedom.

Table II shows the results of the comparison for the recognition task. Two different vehicles (275 different examples) were added to the test set creating what is called the confuser class [38]. Now the problem becomes much more difficult because we are measuring how representative the discriminant function is for the given class in an extended operating environment. With the test set data we check the generalization performance in areas of the input space close the classes, but this still leaves out many unexplored regions of the space where the classifier will provide a class assignment. Ideally the response of the classifier to other vehicles not present in the training set (which reside away from the training data) should be zero. But the conventional training does not enforce this. The problem becomes a blend of detection and classification, which we call recognition. We have to establish a detection threshold for the comparison (in fact a receiver operating characteristic would be more appropriate). Here the realistic probability of detection of Pd = 0.9 is chosen, and the results presented in Table II. In this task, a good classifier will produce low misclassification error and reject as many confusers as possible. We see that the ED-QMI has comparable performance to the SVM machine. The average classifier error rate is slightly better than the SVM but the rejection rate to confusers is slightly worse (64 versus 68%).

Table II. Classification error (%) and confuser rejection (%) for Pd=0.9

	BMP2	BTR70	T72	Average	Confuser
OH	3.87	0.93	2.28	2.76	48
SVM	3.55	0.93	0.98	2.07	68
ED-QMI	3.95	0.75	0.95	1.88	64
Perceptron	3.87	1.87	6.19	4.56	22

The rejection to confusers is highly dependent upon the type of discriminant function that the network topology can create. We [38] (and others [14]) have shown that the most suitable discriminant function for the task of rejection is a local discriminant function. Global discriminant functions such as hyperplanes produce with high probability large responses in areas of the input space away from the class clusters, while local discriminant functions naturally bound the class. This partially explains the difference between the OH and the SVM since they are trained with the same algorithm, except that one creates linear discriminant functions in the input space (OH) while SVMs create local discriminant in pattern space. The SVM outperforms the OH for confuser rejection. But notice that the ED-QMI also uses linear discriminant functions in the input space while its performance is much closer to the SVM than to the OH classifier. Hence, we conclude that the mutual information training is creating discriminant functions that fit tightly the class cluster, comparable to the best classifiers. As Table II clearly shows, the perceptron trained with the delta rule totally breaks down for the task of recognition (it can only reject 22% of the confuser vehicles). This shows that MSE training places discriminant functions to meet the training set criterion but do not guarantee a good match to the data clusters in the input space.

8 Conclusion

We develop in this paper a framework for information theoretic learning that does not require the selection of data models. Hence, the learning machine is able to learn directly from the data just like the conventional MSE criterion. Under this framework, entropy and mutual information manipulations at the output of any linear or nonlinear mappers are possible. Moreover, the same algorithm can be used for supervised and unsupervised learning. We utilize the concept of Renyi's Quadratic entropy along with the Parzen window pdf estimation to develop an easily implementable entropy estimator based on information potential. With this estimator of entropy in the output space of a mapper we can adapt parameters using information force backpropagation. Using the Cauchy-Schwartz and the Euclidean distances instead of Kulback-Leibler divergence we are able to extend the method to mutual information. We applied the novel algorithm to two applications: blind source separation (an unsupervised problem) and to automatic target recognition in SAR. In both cases we showed that the new algorithm performed at the same level of the best algorithms available. This shows the potential of the new technique. Present work is addressing details in the training such as the kernel size, and the effect of the number of dimensions of the output space in the performance. Generalization is also being investigated and compared with that of the MLP and SVMs. We are also studying the statistical properties of the new estimators for entropy and mutual information. The algorithm developed here are $O(N^2)$ where N is the number of samples in the training set. This seems to be an intrinsic limitation since Renyi's quadratic entropy is computed from the interactions of pairs of data samples. On one hand this criterion uses more information about the input data (a data set with N samples has Nx(N-1)/2 different pairs) but it takes longer to compute. Still the algorithm seems to train smoothly and it is not very sensitive to training parameters.

Acknowledgments:

We would like to acknowledge the contributions of Dave Gross and Jim Schmidt from Veridian for the XPATCH simulations.

References

H. Akaike. "A new look at the statistical model identification". *IEEE Transactions on Automatic Control*, 19:716-723, 1974.

J. Anlauf and M. Biehl. "The Adatron: an adaptive perceptron algorithm". *Europhysics Letters*, 10(7): 687-692, 1989.

L. Breiman. *Bagging predictors*. Technical report No.421, University of California, Berkeley, 1994.

R. Chellappa, C. Wilson, and S. Sirohey. "Human and Machine Recognition of Faces: A Survey". *Proceedings of the IEEE*, 83(5), 1995.

C. Burges. "A tutorial on support vector machines for pattern recognition". To appear in *Data Mining and Knowledge Discovery*, 1998.

C. Cortes and V. Vapnik. "Support vector networks". *Machine Learning*, 20: 273-297, 1995.

R. Courant and D. Hilbert. *Methods of mathematical physics*, Interscience, 1953.

J. Fisher and J.C. Principe. "Recent advances to nonlinear MACE filters". *Optical Engineering*, 36(10): 2697-2709, 1998.

R. Fletcher. *Practical methods of optimization*. Great Britain: John Wiley and Sons, Inc., 2nd edition, 1987.

Y. Freund and R. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In *Proceedings of the 2nd European Conference on Computational Learning Theory*, 1995.

T. Frieβ, N. Cristianini and C. Campbell. "The kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines". In *Machine Learning: Proceedinds of the 15th International Conference*, Shavlik, J. ed., Morgan Kaufmann Publishers, San Francisco, CA, 1998.

T. Frieβ. "Support vector neural networks: the kernel Adatron with bias and soft-margin". Research report, University of Sheffield, UK, 1998.

K. Fukunaga. *Statistical pattern recognition*. 2nd ed. San Diego, CA:Academic Press.

H. Gish and M. Schimdt. "Text-independent speaker identificatio". *IEEE Signal Processing Magazine*, 11: 18-32, 1994.

F. Girosi. "An equivalence between sparse approximation and support vector machines". *Neural Computation*, 10(6): 1455-1480, 1998.

M. Gori and F. Scarselli. "Are multilayer perceptrons adequate for pattern recognition and verification?". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11): 1121-1132, 1998.

S. Haykin. *Neural networks: A comprehensive foundation*, Englewood, NJ: Macmillan College Company, Inc, 1994.

C. W. Helstrom, *Statistical theory of signal detection.* 2^{nd} edition, Pergamon Press Inc., 1968.

B. Juang and S. Katagiri. "Discriminative learning for minimum error classification." *IEEE Transactions on Signal Processing*, 40(12): 3043-3054, 1992.

S. H. Lin, S. Y. Kung, and L.J. Lin. "Face recognition/detection by probabilistic decision-based neural network". *IEEE Transactions on Neural Networks*, 8(1): 114-132, 1997.

R. Lippmann. "An introduction to computing with neural nets". *IEEE ASSP Magazine*, pp.4-22, 1987..

N. Nilsson, *Learning Machines: foundations of trainable pattern-classifying systems*, McGraw-Hill, Inc., 1965.

L. Novak, G. Owirka, W. Brower and A. Weaver. "The automatic target recognition system in SAIP", The Lincoln Lab Journal, 10(2), 187-202, 1997.

J. Principe, A. Radisavljevic, J. Fisher, and L. Novak. "Target prescreening based on a quadratic Gamma discriminator". *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):706-715, 1998a.

J. Principe, Q. Zhao and D. Xu. "A novel ATR classifier exploiting pose information". In *Proceedings of Image Understanding Workshop*, pp.833-836, Monterey, CA., Nov. 1998b.

M. Priestley, *Spectral analysis and time series*, New York: Academic Press, 1981.

J. Rissanen. "Modeling by shortest data description". *Automatica*, 14: 465-471, 1978.

J. Rissanen, *Stochastic complexity in statistical inquiry*, Singapore: World Scientific, 1989.

F. Rosenblatt. The Perceptron: "A probabilistic model for information storage and organization in the brain". *Psychological Review*, 65: 386-408, 1958.

D. Rumelhart, G. Hinton, and R. Williams. "Learning internal representations by error propagation". In *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, 1, Chapter 8, Cambridge, MA: MIT Press, 1986.

R. Schapire, Y. Freund, P. Bartlett, and W. Lee. "Boosting the margin: A new explanation for the effectiveness of voting methods". To appear in *Annals of Statistics*, 1998.

M. Schimdt. "Identifying speaker with support vector networks". In *Interface '96 Proceedings*, Sydney, 1996.

A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems.* Washington, D.C.: W.H. Winston, 1977.

H. L. Van Trees. *Satellite communications*, Wiley, 1979.

V. Vapnik. *The nature of statistical learning theory.* New York: Springer-Verlag, Inc., 1995.

V. Velten, T. Ross, J. Mossing, S. Worrell, and M. Bryant. "Standard SAR ATR Evaluation Experiments using the MSTAR Public Release Data Set". Research Report, Wright State University, 1998.

A.S. Weigend. "Generalization by weight elimination with application to forecasting". In *Adavances in Neural Information processing Systems,* 3:875-882, 1991.

Q. Zhao and Z. Bao. "Radar target recognition using a radial basis function neural network". *Neural Networks,* 9(4), pp.709-720, 1996.

Q. Zhao, D.X. Xu, and J. Principe (1998). "Pose estimation of SAR automatic targetrecognition." In *Proceedings of Image Understanding Workshop,* Monterey, CA., Nov. 1998, pp.827-832.

Q. Zhao, and J. Principe. "From hyperplanes to large margin classifiers: Applications to SAR ATR". Proceeding of *SPIE's 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls,* Vol. 3718, Orlando, FL., 1999.

Amari S., Chichocki A., Yang H., "A new learning algorithm for blind source separation", In Advances of Information Processing Systems 8, pp 757-763, 1996.

Barlow H., Unsupervised learning, Neural Computation, vol 1, 295-311, 1989.

Bell A. and Sejnowski T." An information-maximization approach to blind separation and blind deconvolution", Neural Computation, 7:1129-1159, 1995.

Cover T. and Thomas J., "Elements of Information Theory", Wiley, 1991.

Deco G. and Obradovic D., "An Information-Theoretic Approach to Neural Computing", New York, Springer, 1996

Diamantaras K., and Kung S., "Principal Component Neural Networks: Theory and Applications, Wiley, 1996.

Duda, R.O., Hart P.E. "Pattern Classification and Scene Analysis", John Wiley & Sons, 1973.

Fano R., "Transmission of information", MIT Press, 1961.

Fisher J. W. III "Nonlinear Extensions to the Minimum Average Correlation Energy Filter" Ph.D dissertation, Dept. of ECE, University of Florida, 1997

Foldiak P., "Adaptive network for optimal linear feature extraction", IEEE Int. Joint Conf. Neural Net., vol 1, 401-405, 1989.

Friess T., Support vector neural networks: the kernel Adatron with bias and soft margin", Research report, U. of Sheffield, UK, 1998.

Grassberger I., and Proccacia I., "Measuring the strangeness of strange attractors", Physica D, vol 9, 189-208, 1983.

Golub G. and Van Loan F., "Matrix computations", Johns Hopkins Press, 1989.

Gori M and Scarselli F., "Are multilayer perceptrons adequate for pattern recognition and verification?, IEEE Trans. Pattern Analysis and Machine. Intell. 20(11):1121-1132, 1998.

Haykin S., "Neural Networks, A Comprehensive Foundation", Macmillan Publishing Company, 1998.

Haykin S., "Adaptive Filter Theory", Prentice Hall, 1986.

Jaynes E., "In formation theory and statistical mechanics", Physical Review, vol 106, 620-630, 1957.

Kapur, J.N. "Measures of Information and Their Applications". John Wiley & Sons. 1994

Kohonen T., "Self Organizing Maps", Springer Verlag, 1997.

Linsker R. "An application of the principle of maximum information preservation to linear systems", in Advances in Neural Information Processing Systems 1, Morgan-Kaufman, pp 485-494, 1988.

MSTAR (public) Targets, CDROM, Veda Inc. Ohio, 1997.

Novak L., Owirka G., Netishen C., "Performance of a high resolution polarimetric SAR automatic target recognition system, Lincoln Lab. J., 6, 1, 11-23, 1993.

Olshausen B. and Fields D., "Sparse coding with an overcomplete basis set: a strategy employed by V1", Vision research, vol 37, 3311-3325, 1997.

Parzen, E. "On the estimation of a probability density function and the mode", Ann. Math. Stat. 33, p1065, 1962.

Plumbley M., Fallside F., "An information theoretic approach to unsupervised networks", Int. J. Conf. on Neural Nets, vol 2, p 598, Washington, DC, 1989.

Principe J., "From linear adaptive to information filtering", Key note address, IEEE Workshop Neural Nets for Sig. Proc., Cambridge, England, August 1998.

Principe J., Xu D., Fisher J., "Information theoretic learning", in Unsupervised Adaptive Filtering, Ed. Haykin, Wiley, 2000 (in press).

Renyi, A. "Some Fundamental Questions of Information Theory", Selected Papers of Alfred Renyi, Vol.2, Akademic Kiado, Budapest, 1976.

Rumelhart, D.E., Hinton, G.E. and Williams, J.R. "Learning representations by back-propagating errors", Nature (London), 323, pp533-536, 1986.

Shannon C. and Weaver W., "The mathematical theory of communication", University of Illinois Press, 1949.

Wu H-C, principe J., Novel Quadratic Entropy measures and their application to blind source separation/extraction, accepted in IEEE Workshop Neural Networks Sig. Proc. 1999

Vapnik V., "Statistical Learning theory", Wiley, 1998.

Velten V., Ross T. Mossing J., Worrell S., Bryant M., "standard SAR/ATR evaluation experiments using the MSTAR public release data set", Research Report, Wright State U., 1998.

Viola P., Schraudolph N., Sejnowski T., "Empirical entropy manipulation for real-world problems", Proc. Neural Info. Proc. Sys. (NIPS 8) Conf., 851-857, 1995.

Xu D., Principe J., Fisher J. and Wu H-C. "A Novel Measure for Independent Component Analysis (ICA)" Proc. ICASSP'98, vol II, 1161-1164, 1998.

Xu D., Fisher J., Principe J., "Mutual Information approach to pose estimation", Proc. SPIE, vol 3370, Algorithms for synthetic aperture radar imagery V, 218-229, 1998.

Xu D., "Energy, Entropy and Information Potential for Neural Computation", Ph.D. dissertation, U. of Florida, 1999.

Zhao Q. and J. Principe, "From hyperplanes to large margin classifiers: Appllications to SAR/ATR", In Proc. SPIE 13th Annual Int. Sym. Aerospace/Defense Sensing, Simulation and Control, Vol 3718, 1999.

Abu-Mostafa; Y.S. (1995) "Hints". Neural Computation. 7:639-671.

Barron, A.R. (1994) "Approximation and estimation bounds for artificial neural networks". Machine Learning. 14:115-133.

Bishop, C.M. (1995). Neural networks for patter recognition. New York: Oxford University Press Inc.

Girosi, F., Poggio, T. & Caprile B. (1991) "Extensions of a theory of networks for approximation and learning: outliers and negative examples". In R.P. Lippmann, J.E. Moody and D. S. Touretzky (eds.) Advances in Neural Information Processing Systems 3, pp. 750-756. San Mateo, CA: Morgan Kaufmann.

Helstrom, C.W. (1968). Statistical theory of signal detection. 2nd edition, Pergamon Press Inc.

Holmstrom, L. & Koistinen, P. (1992) "Using additive noise in back-propagation training". IEEE Trans. Neural Networks. 3(11):24-38.

Nilsson, N. (1965). Learning Machines: foundations of trainable pattern-classifying systems. McGraw-Hill, Inc.

Niyogi, P. & Girosi, F. (1996) "On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions". Neural Computation. 8:819-842.

Niyogi, P. & Girosi, F. & Poggio, T. (1998) "Incorporating prior information in machine learning by creating virtual examples". Proceedings of the IEEE, 86(11):2196-2209.

Novak, L. & Owirka,G. & Brower, W. & Weaver, A. (1997) "The automatic target recognition system in SAIP". The Lincoln Lab Journal: 10(2), 187-202.

Principe, J. & Radisavljevic, A. & Fisher, J. & Novak, L. (1998) "Target prescreening based on a quadratic Gamma discriminator". IEEE Transactions on Aerospace and Electronic Systems, 34(3):706-715.

Simard, P. & LeCun, Y. & Denker, J. (1993) "Efficient pattern recognition using a new transformation distance". In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), Advances in Neural Information Processing Systems 5, pp. 50-58. San Mateo, CA: Morgan Kaufman

Tikhonov, A. and Arsenin,V. (1977). Solutions of ill-posed problems. Washington, D.C.: W.H. Winston.

Vapnik, V. (1995). The nature of statistical learning theory. New York: Springer-Verlag, Inc.

Webb, A.R. (1994) "Functional approximation by feed-forward networks: a least-squares approach to generalization". IEEE Transactions on Neural Networks, 5(3):363-371.

Zhao, Q. & Principe, J. (1999) "From hyperplanes to large margin classifiers: Applications to SAR ATR". In Proceeding of SPIE's 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Vol.3718.